Administration des réseaux TCP/IP, d'IPMP et des tunnels IP dans Oracle® Solaris 11.2

Référence: E53799
Juillet 2014

ORACLE

Table des matières

Utilisation de la présente documentation

- **Présentation** : décrit les tâches relatives à l'administration de réseaux TCP / IP et les tunnels IP IPMP dans le système d'exploitation Oracle Solaris (OS).
- **Public visé** : administrateurs système.
- **Connaissances requises** : expérience avancée dans l'administration avec d'administration réseau TCP / IP, y compris les fonctions réseau avancées, les réseaux tels que et IP IPMP tunnels IP.

Bibliothèque de la documentation des produits

Les informations de dernière minute et les problèmes connus pour ce produit sont inclus dans la bibliothèque de documentation accessible à l'adresse : http://www.oracle.com/pls/topic/lookup?ctx=E56338.

Accès aux services de support Oracle

Les clients Oracle ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info ou le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs si vous êtes malentendant.

Commentaires en retour

Faites part de vos commentaires sur cette documentation à la page suivante : http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPITRE 1

Administration des réseaux TCP/IP

Ce chapitre décrit comment gérer le protocole TCP / IP sur les systèmes sur lesquels Oracle Solaris installé. L'exécution des tâches présentées dans ce chapitre nécessite l'installation d'un réseau TCP/IP opérationnel sur votre site (IPv4 uniquement ou IPv4/IPv6 double pile).

Pour plus d'informations sur la planification du développement de votre réseau, reportez-vous à " Planification du développement du réseau dans Oracle Solaris 11.2 "

Pour les tâches de configuration réseau, reportez-vous à " Configuration et administration des composants réseau dans Oracle Solaris 11.2 ".

Pour obtenir des informations générales sur le dépannage dans les réseaux TCP / IP, reportez-vous à " Dépannage des problèmes d'administration réseau dans Oracle Solaris 11.2 ".

Ce chapitre aborde les sujets suivants :

La section Personnalisation des propriétés TCP / IP

Vous utilisez la commande `ipadm` pour configuration de la majorité des propriétés TCP/IP, également appelées paramètres *réglables*. La commande `ipadm` remplace la commande `ndd`

comme premier outil de définition de paramètres réglables. Pour plus d'informations sur ces modifications, reportez à “ Comparaison de la commande ndd et de la commande ipadm ” du manuel “ Transition d’Oracle Solaris 10 vers Oracle Solaris 11.2 ”.

Les propriétés TCP/IP peuvent être basées sur l'interface ou globales, et les propriétés peuvent être appliquées à une interface spécifique ou au niveau global à toutes les interfaces à l'intérieur d'une zone. Les propriétés globales peuvent avoir des valeurs différentes dans plusieurs zones non globales. Pour une liste complète des propriétés de protocole, reportez-vous à la page de manuel For a complete list of the supported protocol properties, see the `ipadm`(1M)..

En règle générale, les paramètres par défaut du protocole Internet TCP/IP s'avèrent suffisants pour que le réseau fonctionne. Toutefois, si ces valeurs ne sont pas suffisantes pour votre topologie de réseau particulière, vous pouvez personnaliser les trois propriétés à l'aide des trois sous-commandes `ipadm` suivantes :

- `ipadm show-prop -p`*propertyprotocol* : affiche les propriétés d'un protocole et ses valeurs en cours. Si vous n'utilisez pas l'option -p*property*, toutes les propriétés du protocole sont répertoriées. Si vous ne spécifiez pas de protocole, toutes les propriétés de toutes les liaisons de données s'affichent.
- `ipadm set-prop -p` *property=value protocol* : affecte une valeur à la propriété du protocole.
  - Pour affecter plusieurs valeurs à une propriété d'un protocole, utilisez la syntaxe suivante :

    `ipadm set-prop [-t] -p` *property=value*`[,...] protocol`
  - Pour supprimer une valeur d'un ensemble de valeurs pour une propriété, utilisez le qualificatif -= comme suit :

    `ipadm set-prop -p` *property*`-=`*value2*
- Réinitialisez une propriété de protocole spécifique ses valeurs par défaut comme suit :

  `ipadm reset-prop -p` *property protocol*

Pour plus d'informations sur la personnalisation, reportez-vous à “ Personnalisation des propriétés et des adresses des interfaces IP ” du manuel “ Configuration et administration des composants réseau dans Oracle Solaris 11.2 ”.

Pour plus d'informations sur la personnalisation, reportez-vous à “ Personnalisation des propriétés des adresses IP ” du manuel “ Configuration et administration des composants réseau dans Oracle Solaris 11.2 ”.

Activation globale du transfert de paquets

Lorsque vous activez le transfert sur une interface IP à l'aide de la commande`ipadm set-ifprop`, il n'est activé que pour celle-ci, et le transfert sur toutes les autres interfaces ne change pas. La définition du transfert de paquets au niveau de chaque propriété d'interface IP vous

permet d'implémenter cette fonction de manière sélective sur certaines interfaces du système. Pour plus d'informations, reportez-vous à " Activation de la transmission de paquets " du manuel " Configuration et administration des composants réseau dans Oracle Solaris 11.2 ".

Pour activer le transfert de paquets sur un système entier, quel que soit le nombre de protocole IP des interfaces, utilisez la propriété `protocol`. La propriété `protocol` est la propriété IP générale utilisée pour gérer le transfert avec le même nom de propriété que celui utilisé pour gérer le transfert sur les interfaces IP individuelles. Comme vous pouvez activer le transfert des protocoles IPv4 ou IPv6 (ou les deux), vous devez gérer chaque protocole individuellement.

Par exemple, vous devez activer la transmission de paquets pour l'ensemble du trafic IPv4 et IPv6 sur le système comme suit :

```
# ipadm show-prop -p forwarding ip
PROTO   PROPERTY     PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv4    forwarding   rw     off       --           off       on,off
ipv6    forwarding   rw     off       --           off       on,off

# ipadm set-prop -p forwarding=on ipv4
# ipadm set-prop -p forwarding=on ipv6

# ipadm show-prop -p forwarding ip
PROTO   PROPERTY     PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv4    forwarding   rw     on        on           off       on,off
ipv6    forwarding   rw     on        on           off       on,off
```

Remarque - La propriété `forwarding` de l'une des interfaces IP ou des protocoles n'est pas exclusive. Vous pouvez définir la propriété pour l'interface et le protocole en même temps. Par exemple, vous pouvez activer le transfert de paquets globalement sur le protocole, puis personnaliser le transfert de paquets pour chaque interface IP du système. Par conséquent, même s'il est activé globalement, vous avez toujours la possibilité de gérer le transfert de paquets de façon sélective sur la base d'une interface

Administration de la sélection des adresses par défaut

Oracle Solaris permet à une interface unique de disposer de plusieurs adresses IP. Par exemple, certaines fonctionnalités telles que la fonctionnalité IPMP (multiacheminement sur réseau IP) permettent la connexion de plusieurs cartes d'interface réseau (NIC, network interface card) sur la même couche de liaison IP. Cette liaison peut être associée à une ou plusieurs adresses IP. Les interfaces des systèmes IPv6 possèdent également une adresse IPv6 lien-local, au moins une adresse de routage IPv6 ainsi qu'une adresse IPv4 pour au moins une interface.

Lorsque le système génère une transaction, une application envoie un appel vers le socket `getaddrinfo`. Le socket `getaddrinfo` détecte les adresses possibles en cours d'utilisation sur le système de destination. Le noyau établit alors l'ordre de priorité de cette liste afin de déterminer la destination appropriée pour le paquet. Ce processus est appelé *classement des adresses de*

destination. Le noyau Oracle Solaris sélectionne le format approprié pour l'adresse source en fonction de l'adresse de destination déterminée pour le paquet. Ce processus est appelé *sélection des adresses*. Pour plus d'informations sur le classement des adresses de destination, reportez-vous à la page de manuel `getaddrinfo`(3SOCKET).

Le processus de sélection des adresses par défaut doit s'effectuer sur les systèmes IPv4 uniquement ainsi que sur les systèmes double pile IPv4/IPv6. Dans la plupart des cas, il n'est pas nécessaire de modifier les mécanismes de sélection des adresses par défaut. Toutefois, vous devrez peut-être modifier l'ordre de priorité des formats d'adresse de manière à prendre en charge la fonctionnalité IPMP ou à préférer les formats d'adresse 6to4, par exemple.

Description du fichier de configuration `/etc/inet/ipaddrsel.conf`

Le fichier `/etc/inet/ipaddrsel.conf` contient la table des règles de sélection d'adresse IPv6 par défaut. Si vous avez activé le protocole IPv6 lors de l'installation Oracle Solaris, ce fichier contient les éléments présentés dans Tableau 1-1, “Tableau des règles de sélection des adresses IPv6 par défaut”.

Vous pouvez modifier le contenu de `/etc/inet/ipaddrsel.conf`. Toutefois, cette opération n'est pas recommandée. Si cela s'avère nécessaire, reportez-vous à la procédure décrite à la section “Administration de la table des règles de sélection d'adresses IPv6” à la page 14. Pour plus d'informations sur le fichier `ippaddrsel.conf`, reportez-vous à la section “Raisons pour lesquelles le tableau des règles de sélection d'adresses IPv6 doit être modifié” à la page 13 ainsi qu'à la page de manuel `ipaddrsel.conf`(4).

Description de la commande `ipaddrsel`

La commande `ipaddrsel` permet de modifier le tableau des règles de sélection des adresses IPv6 par défaut.

Le noyau Oracle Solaris utilise la table des règles de sélection des adresses IPv6 par défaut pour le classement des adresses de destination et la sélection des adresses sources pour les en-têtes de paquet IPv6. Le fichier `/etc/inet/ipaddrsel.conf` contient ce tableau de règles.

Le tableau suivant répertorie les formats d'adresse par défaut ainsi que les priorités de chacune telles qu'elles doivent figurer dans le tableau de règles. Vous pouvez rechercher des informations techniques sur la sélection d'adresses IPv6 dans la page de manuel `inet6`(7P).

TABLEAU 1-1 Tableau des règles de sélection des adresses IPv6 par défaut

Préfixe	Ordre de priorité	Définition
`::1/128`	50	Loopback
`::/0`	40	Par défaut :
`2002::/16`	30	6to4
`::/96`	20	Compatible IPv4
`::ffff:0:0/96`	10	IPv4

Dans ce tableau, les préfixes IPv6 (`::1/128` et `::/0`) ont la priorité sur les adresses 6to4 (`2002::/16`) et les adresses IPv4 (`::/96` et `::ffff:0:0/96`). Par conséquent, le noyau choisit par défaut l'adresse IPv6 globale de l'interface pour les paquets envoyés vers une autre destination IPv6. L'adresse IPv4 de l'interface est moins prioritaire, notamment pour les paquets envoyés vers une destination IPv6. Etant donné l'adresse IPv6 source sélectionnée, le noyau utilise également le format IPv6 pour l'adresse de destination.

Raisons pour lesquelles le tableau des règles de sélection d'adresses IPv6 doit être modifié

En règle générale, le tableau des règles de sélection d'adresses IPv6 par défaut n'a pas besoin d'être modifié. En cas de modification nécessaire, exécutez la commande `ipaddrsel`.

Les situations suivantes nécessitent une modification du tableau :

- Si le système possède une interface qui est utilisée pour un tunnel 6to4, vous pouvez définir une priorité plus élevée pour les adresses 6to4.
- Si vous souhaitez qu'une adresse source particulière communique avec une adresse de destination particulière, vous pouvez ajouter ces adresses au tableau de règles. Ensuite, vous pouvez les marquer comme vos adresses préférées à l'aide de la commande `ipadm`. Pour plus d'informations, reportez-vous à la page de manuel `ipadm`(1M).
- Si vous voulez que les adresses IPv4 aient la priorité sur les adresses IPv6, vous pouvez remplacer la priorité de `::ffff:0:0/96` par un chiffre plus élevé.
- Si vous devez assigner une priorité plus élevée à des adresses désapprouvées, vous pouvez ajouter ces adresses au tableau de règles. Prenons l'exemple des adresses de site locales, actuellement désapprouvées sur le réseau IPv6. Ces adresses possèdent le préfixe `fec0::/10` . Vous pouvez modifier le tableau de règles afin de définir une priorité plus élevée pour ces adresses.

Pour plus d'informations sur la commande `ldapaddent`, reportez-vous à la page de manuel `ipaddrsel`(1M).

Administration de la table des règles de sélection d'adresses IPv6

La procédure suivante explique comment modifier la table des règles de sélection des adresses. Pour obtenir des informations conceptuelles sur la sélection des adresses IPv6 par défaut, reportez-vous à Description de la commande ipaddrsel.

Attention - Ne modifiez pas la table à l'exception des règles de sélection des adresses IPv6 sur la base des motifs qui sont fournis dans la procédure suivante. Les erreurs de définition de la table des règles risquent d'entraîner des problèmes de fonctionnement du réseau. Par ailleurs, veillez à enregistrer une copie de sauvegarde de la table des règles, comme illustré dans cette procédure.

1. **Connexion en tant qu'administrateur.**

2. **Révision du tableau de règles de sélection des adresses IPv6 actuel.**

```
# ipaddrsel
# Prefix                 Precedence Label
::1/128                          50 Loopback
::/0                             40 Default
2002::/16                        30 6to4
::/96                            20 IPv4_Compatible
::ffff:0.0.0.0/96                10 IPv4
```

3. **Effectuez une copie de la table des règles de sélection d'adresses par défaut.**

```
# cp /etc/inet/ipaddrsel.conf /etc/inet/ipaddrsel.conf.orig
```

4. **Ajouter toutes les personnalisations dans le fichier `/etc/inet/ipaddrsel.conf`.**

```
# pfedit /etc/inet/ipaddrsel.conf
```

Utilisez la syntaxe suivante pour les entrées de fichier `/etc/inet/ipaddrsel` :

prefix/prefix-length precedence label [# comment]

Reportez-vous à Exemple 1-1, “La modification de la table des règles de sélection d'adresses Pv6 par défaut” pour les modifications plus communes que vous pouvez effectuer.

5. **Chargez la table de règles modifiée dans le noyau.**

```
# ipaddrsel -f /etc/inet/ipaddrsel.conf
```

6. **Si la table des règles modifiée génère des erreurs, restaurez la table des règles de sélection des adresses IPv6 par défaut.**

```
# ipaddrsel -d
```

Exemple 1-1 La modification de la table des règles de sélection d'adresses Pv6 par défaut

Les exemples ci-dessous illustrent les modifications susceptibles d'être apportées le plus souvent à la table des règles :

- Définition des adresses 6to4 sur la priorité la plus élevée :

```
2002::/16                       50 6to4
::1/128                         45 Loopback
```

  Le format d'adresse 6to4 dispose dorénavant de la plus haute priorité (50). Loopback, qui disposait auparavant d'une priorité de 50, dispose dorénavant d'une priorité de 45. Les autres formats d'adresse restent inchangés.

- Définition d'une adresse source spécifique pour les communications avec une adresse de destination donnée :

```
::1/128                         50 Loopback
2001:1111:1111::1/128           40 ClientNet
2001:2222:2222::/48             40 ClientNet
::/0                            40 Default
```

  Ce type de configuration s'utilise notamment pour les hôtes associés à une seule interface physique. Dans cet exemple, l'adresse source `2001:1111:1111::1/128` est définie en tant qu'adresse prioritaire pour les paquets adressés aux destinations du réseau `2001:2222:2222::/48`. L'adresse source `2001:1111:1111::1/128` est associée à la priorité 40, priorité supérieure à celle des autres formats d'adresse configurés pour l'interface.

- Préférence des adresses IPv4 par rapport aux adresses IPv6 :

```
::ffff:0.0.0.0/96               60 IPv4
::1/128                         50 Loopback
.
.
```

  La priorité par défaut du format IPv4 `::ffff:0.0.0.0/96` passe de 10 à 60, soit la priorité la plus élevée de la table.

▼ Modification de la table des règles de sélection des adresses IPv6 pour la session en cours uniquement

Les modifications apportées au fichier `/etc/inet/ipaddrsel.conf` sont conservées lors des sessions suivantes. Si vous souhaitez modifier la table des règles uniquement pour la session en cours, effectuez la procédure suivante.

1. **Connexion en tant qu'administrateur.**

2. **Tout d'abord copiez le contenu du fichier `/etc/inet/ipaddrsel` vers *filename*, où *filename*désigne n'importe quel nom de votre choix.**

   ```
   # cp /etc/inet/ipaddrsel filename
   ```

3. **Utilisez la commande `pfedit` pour éditer le tableau des règles dans *filename* pour vos spécifications.**

4. **Chargez la table de règles modifiée dans le noyau.**

   ```
   # ipaddrsel -f filename
   ```

   Le noyau utilise la nouvelle table des règles jusqu'au prochain redémarrage du système.

Administration des services de couche transport

Les protocoles de couche de transport suivants constituent une partie standard d'Oracle Solaris : Transmission Control Protocol (TCP), Stream Control Transmission Protocol (SCTP), et User Datagram Protocol (UDP). Généralement, ces protocoles fonctionnent correctement sans que l'utilisateur ait à intervenir. Toutefois, dans certaines conditions, vous serez peut-être amené à consigner ou modifier des services exécutés via les protocoles de couche transport. Dans ce cas, vous devez modifier les profils de ces services à l'aide de SMF (Service Management Facility) des commandes.

Le démon `inetd` est chargé de lancer les services Internet standard lors de l'initialisation d'un système. Ces services incluent les applications utilisant les protocoles de couche transport TCP, SCTP ou UDP. Vous pouvez modifier les services Internet existants ou ajouter de nouveaux services à l'aide des commandes SMF.

Opérations impliquant les protocoles de couche transport :

- Définition d'un port privilégié
- Implémentation du contrôle de congestion du trafic
- Journalisation de toutes les connexions TCP entrantes
- Ajout de services faisant appel à un protocole de couche transport, utilisant SCTP comme exemple
- Configuration des wrappers TCP dans le cadre du contrôle d'accès
- Modification de la taille du tampon de réception TCP

Pour plus d'informations, reportez-vous à“ Modification des services contrôlés par inetd ” du manuel “ Gestion des services système dans Oracle Solaris 11.2 ” et à la page de manuel `inetd`(1M).

Définition d'un port privilégié

Dans les protocoles de transport tels que TCP, UDP et SCTP, les ports `1-1023` sont par défaut des ports privilégiés. Pour établir la liaison avec la liste des ports privilégiés, un processus doit être en cours d'exécution avec les autorisations `root`. Les ports dont le statut est supérieur à `1023` sont, par défaut, sans privilèges. Vous pouvez utiliser la commande `ipadm` pour étendre la plage de ports privilégiés, ou vous pouvez cocher des ports spécifiques dans la plage non privilégié comme ports privilégiés.

Pour gérer la plage de ports privilégiés, vous pouvez personnaliser les propriétés de protocole de transport suivantes :

`smallest_nonpriv_port` Indique une valeur qui indique le début de la plage de numéros de port non privilégié, qui sont les ports avec lesquels les utilisateurs standard peuvent créer des liaisons. Vous pouvez définir des ports individuels sont plage au sein de la non privilégié comme ports privilégiés. Exécutez la commande `ipadm show-prop` pour afficher les valeurs de la propriété.

`extra_priv_ports` Spécifie les ports en dehors de la plage sont également doté de privilèges doté de privilèges. Exécutez la sous-commande `ipadm set-prop` pour spécifier les ports dont vous souhaitez restreindre l'accès. Vous pouvez attribuer plusieurs valeurs à la propriété.

A titre d'exemple, supposons que vous souhaitez définir les ports TCP `3001` et `3050` comme ports privilégiés avec accès réservé uniquement au rôle `root`. La propriété `smallest_nonpriv_port` indique que `1024` est le numéro de port le plus bas des ports non privilégiés. Par conséquent, vous pouvez modifier les ports désignés `3001` et `3050` comme suit :

```
# ipadm show-prop -p smallest_nonpriv_port tcp
PROTO PROPERTY                PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
tcp   smallest_nonpriv_port   rw     1024      --           1024      1024-32768

# ipadm show-prop -p extra_priv_ports tcp
PROTO   PROPERTY           PERM   CURRENT     PERSISTENT   DEFAULT     POSSIBLE
tcp     extra_priv_ports   rw     2049,4045   --           2049,4045   1-65535

# ipadm set-prop -p extra_priv_ports+=3001 tcp
# ipadm set-prop -p extra_priv_ports+=3050 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO   PROPERTY           PERM   CURRENT     PERSISTENT   DEFAULT     POSSIBLE
tcp     extra_priv_ports   rw     2049,4045   3001,3050    2049,4045   1-65535
                                  3001,3050
```

Vous pouvez supprimer un port privilégié, par exemple `4045`, comme suit :

```
# ipadm set-prop -p extra_priv_ports-=4045 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO   PROPERTY           PERM   CURRENT     PERSISTENT   DEFAULT     POSSIBLE
```

```
tcp     extra_priv_ports   rw     2049,3001   3001,3050    2049,4045   1-65535
                                  3050
```

Implémentation du contrôle de congestion du trafic

La congestion du réseau se traduit généralement par des dépassements du tampon du routeur, lorsque les noeuds envoient plus de paquets que le réseau ne peut en gérer. Différents algorithmes empêchent une congestion du trafic à travers la mise en place de contrôles sur les systèmes d'envoi. Ces algorithmes sont pris en charge par Oracle Solaris et peuvent être facilement ajoutés ou directement rattaché à l'OS, comme indiqué dans le tableau suivant qui décrit les algorithmes pris en charge intégrée.

Algorithme	Nom Oracle Solaris	Description
NewReno	`newreno`	Algorithme par défaut dans Oracle Solaris. Ce mécanisme de contrôle est basé sur une fenêtre de congestion de l'expéditeur et les mécanismes de Slow Start (démarrage lent) et de Congestion Avoidance (anti-congestion).
HighSpeed	`highspeed`	L'une des modifications de NewReno les plus connues et les plus simples pour les réseaux à haut débit.
CUBIC	`cubic`	Actuellement l'algorithme par défaut dans Linux 2.6. Avec cet algorithme, la phase Congestion Avoidance passe d'une augmentation linéaire du fenêtrage à une fonction cubique.
Vegas	`vegas`	Algorithme classique basé sur la temporisation, qui tente de prédire la congestion sans déclencher une perte réelle de paquets.

Pour activer le contrôle de congestion, les propriétés de contrôle TCP suivantes doivent être définies. Bien que ces propriétés concernent le trafic TCP, le mécanisme de contrôle activé par ces propriétés s'applique également au trafic SCTP.

`cong_enabled` Contient une liste d'algorithmes, séparés par des virgules, actuellement opérationnels dans le système. Vous pouvez ajouter ou supprimer des algorithmes pour n'utiliser que ceux qui vous intéressent. Cette propriété peut avoir plusieurs valeurs. Vous devez donc utiliser le qualificatif `+=` ou `-=`, selon le changement que vous souhaitez appliquer.

`cong_default` Utilisé par défaut lorsque des applications aucun algorithme n'est spécifié explicitement dans les options de socket. Actuellement la valeur de la propriété `cong_default` s'applique aux zones globales et non globales.

L'exemple ci-dessous illustre comment ajouter un algorithme de contrôle de congestion au protocole, procédez comme suit :

```
# ipadm set-prop -p cong_enabled+=algorithm tcp
```

Supprimer un algorithme comme suit :

```
# ipadm set-prop -p cong_enabled-=algorithm tcp
```

Remplacer l'algorithme par défaut comme suit :

```
# ipadm set-prop -p cong_default=algorithm tcp
```

Remarque - L'ajout ou la suppression d'algorithmes n'est soumise à aucune règle de séquence. Vous pouvez supprimer un algorithme avant d'en ajouter d'autres à une propriété. Cependant, vous devez toujours définir un algorithme pour la propriété `cong_default`.

L'exemple suivant illustre la manière dont vous pouvez implémenter le contrôle de congestion. Dans l'exemple, l'algorithme par défaut du protocole TCP `newreno` est remplacé par `cubic`. Ensuite, l'algorithme `vegas` est retiré de la liste des algorithmes activés.

```
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY              PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
tcp   extra_priv_ports      rw   2049,4045    --           2049,4045    1-65535

# ipadm show-prop -p cong_default,cong_enabled tcp
PROTO PROPERTY              PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
tcp   cong_default          rw   newreno      --           newreno      newreno,cubic,
                                                                       highspeed,vegas
tcp   cong_enabled          rw   newreno,cubic, newreno,cubic, newreno  newreno,cubic,
                                 highspeed,   highspeed,                highspeed,vegas
                                 vegas        vegas

# ipadm set-prop -p cong_enabled-=vegas tcp
# ipadm set-prop -p cong_default=cubic tcp

# ipadm show-prop -p cong_default,cong_enabled tcp
PROTO PROPERTY              PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
tcp   cong_default          rw   cubic        cubic        newreno      newreno,cubic,
                                                                       highspeed
tcp   cong_enabled          rw   newreno,cubic, newreno,cubic, newreno  newreno,cubic,
                                 highspeed    highspeed                highspeed,vegas
```

Journalisation des adresses IP de toutes les connexions TCP entrantes

Les adresses IP pour toutes les connexions TCP entrantes sont répertoriées par le service `syslog` au niveau et dans l'utilitaire `daemon.notice`. Les informations se trouvent dans `/var/adm/messages` sauf si vous avez modifié la configuration de fichier local `syslog.conf`. Notez

que la modification du fichier `syslog.conf` peut avoir une incidence sur l'emplacement où ces informations sont stockées. Pour de plus amples détails, reportez-vous à la page de manuel `syslog.conf`(4).

Définissez le traçage TCP sur `TRUE` (activé) pour tous les services gérés par le démon `inetd` comme suit :

```
# inetadm -M tcp_trace=TRUE
```

Ajout de services That Use the SCTP Protocol

L'SCTP (Stream Control Transmission Protocol, protocole, échange de clés sur Internet) fournit des services aux protocoles de couche d'application de façon similaire à TCP. Toutefois, SCTP permet la communication entre deux systèmes à multihébergement ou deux systèmes dont l'un est à multihébergement. La connexion SCTP s'appelle une *association*. Dans une association, une application divise les données à transmettre en *plusieurs flux de messages*. Une connexion SCTP peut atteindre les extrémités à l'aide de plusieurs adresses IP, ce qui s'avère particulièrement important dans le cadre d'applications de téléphonie. Les capacités multiréseau de SCTP améliorent la sécurité des sites ayant recours à IP Filter ou IPsec. La page de manuel `sctp`(7P)répertorie les points à prendre en considération au niveau de la sécurité.

▼ Ajout de services utilisant le protocole SCTP

Par défaut, le protocole SCTP fait partie d'Oracle Solaris et ne nécessite aucune configuration supplémentaire. Toutefois, vous devrez peut-être configurer explicitement certains services de couche d'application pour utiliser SCTP. Des exemples sont les applications `echo` et `discard`. La procédure suivante explique comment ajouter un service d'écho qui utilise un socket style SCTP un à un. Vous pouvez utiliser la même procédure et d'ajouter des services pour TCP UDP.

La tâche suivante explique comment ajouter un service SCTP `inet`, qui est géré par SMF le démon de, au référentiel `inetd`. Dans ce cas, la tâche explique comment utiliser les commandes d'ajout du service SMF.

Avant de commencer Avant d'effectuer la procédure suivante, créez un fichier manifeste pour le service. A titre d'exemple, cette procédure s'effectue via un fichier manifeste du service appelé `echoecho.sctp.xml`.

1. **Connectez-vous au système local avec un compte utilisateur disposant de privilèges d'écriture sur les fichiers système.**
2. **Ajoutez la définition du nouveau service au fichier à l'aide de la commande `/etc/servicespfedit`.**

   Reportez-vous à la page de manuel `pfedit`(1M).

Utilisez la syntaxe suivante pour la définition du service:

service-name port/protocol aliases

3. **Accédez au répertoire de stockage du manifeste de service, puis importer le manifeste de service.**

```
# cd dir-name
# svccfg import service-manifest-name
```

Par exemple, vous devez ajouter un service SCTP `echo` à l'aide du manifeste `echo.sctp.xml` qui se trouve dans le répertoire `service.dir` comme suit :

```
# cd service.dir
# svccfg import echo.sctp.xml
```

4. **Assurez-vous que le manifeste de service a été ajouté.**

```
# svcs FMRI
```

Pour l'argument *FMRI*, utilisez le FMRI (Fault Managed Resource Identifier, identificateur de ressources gérées erronées) du manifeste de service.

5. **Dressez la liste des propriétés du service afin d'identifier les modifications à apporter.**

```
# inetadm -l FMRI
```

6. **Activez le nouveau service.**

```
# inetadm -e FMRI
```

7. **Vérifiez que le service est activé.**

Exemple 1-2 Ajout d'un service utilisant le protocole de transport SCTP

L'exemple suivant indique les commandes à utiliser et les entrées de fichier qui doivent être renseignés pour que le service `echo` utilise SCTP.

```
# cat /etc/services
.
.
echo            7/tcp
echo            7/udp
echo             7/sctp

# cd service.dir

# svccfg import echo.sctp.xml

# svcs network/echo*
```

```
STATE          STIME    FMRI
disabled       15:46:44 svc:/network/echo:dgram
disabled       15:46:44 svc:/network/echo:stream
disabled       16:17:00 svc:/network/echo:sctp_stream

# inetadm -l svc:/network/echo:sctp_stream
SCOPE    NAME=VALUE
         name="echo"
         endpoint_type="stream"
         proto="sctp"
         isrpc=FALSE
         wait=FALSE
         exec="/usr/lib/inet/in.echod -s"
         user="root"
default  bind_addr=""
default  bind_fail_max=-1
default  bind_fail_interval=-1
default  max_con_rate=-1
default  max_copies=-1
default  con_rate_offline=-1
default  failrate_cnt=40
default  failrate_interval=60
default  inherit_env=TRUE
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE

# inetadm -e svc:/network/echo:sctp_stream

# inetadm | grep echo
disabled  disabled       svc:/network/echo:stream
disabled  disabled       svc:/network/echo:dgram
enabled   online         svc:/network/echo:sctp_stream
```

La configuration des wrappers TCP

Les wrappers TCP représentent une mesure de sécurité supplémentaire pour les démons de services en s'interposant entre le démon et les requêtes de service entrantes. Les wrappers TCP consignent les réussites et les échecs des tentatives de connexion. En outre, ils offrent un contrôle d'accès en autorisant ou en refusant la connexion en fonction de l'origine de la demande. Vous pouvez utiliser des wrappers TCP de protéger les démons, notamment SSH (Secure Shell le), Telnet et FTP (File Transfer Protocol). L'application `sendmail` peut également utiliser des wrappers TCP, comme décrit à la section “ Prise en charge des wrappers TCP à partir de la version 8.12 de sendmail ” du manuel “ Gestion des services sendmail dans Oracle Solaris 11.2 ”.

▼ Contrôle d'accès aux services TCP à l'aide des wrappers TCP

Le programme `tcpd` implémente des *wrappers TCP*.

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Activez les wrappers TCP.**

   ```
   # inetadm -M tcp_wrappers=TRUE
   ```

3. **Configurer la stratégie de contrôle d'accès des wrappers TCP.**

   Pour obtenir des instructions, reportez-vous à la page de manuel `hosts_access`(3), qui se trouve dans le répertoire `/usr/sfw/man`.

Modification de la taille du tampon de réception TCP

La taille du tampon de réception TCP est défini en utilisant la propriété TCP `recv_buf` qui est de 128 KO par défaut. Or les applications n'utilisent pas toutes la même quantité de bande passante. Par conséquent, le temps de latence de connexion peut exiger un changement de la taille par défaut. Par exemple, l'utilisation de la fonction Secure Shell d'Oracle Solaris pèse excessivement sur la bande passante en raison des processus de somme de contrôle et de chiffrement appliqués au flux de données. Il peut donc être nécessaire d'augmenter la taille de la mémoire tampon. De même, dans le cas d'applications qui effectuent des transferts en bloc, il convient d'ajuster la taille de la mémoire tampon pour assurer une utilisation plus efficace de la bande passante.

Vous pouvez calculer la taille de tampon de réception appropriée en estimant le produit BDP (Bandwidth Delay Product) comme suit : Pour calculer le BDP, multiplier la bande passante disponible allouée par la valeur de la latence de la connexion.

Utilisez la commande `ping -s` *host* pour obtenir la valeur de la latence de la connexion.

La taille du tampon de réception appropriée équivaut plus au moins au produit BDP. Notez cependant que l'utilisation de la bande passante dépend également d'un grand nombre de facteurs. Une infrastructure partagée ou le nombre d'applications et d'utilisateurs qui sollicitent simultanément la bande passante sont des éléments sui peuvent faire varier l'estimation.

Modifier la valeur de la taille de la mémoire tampon de la manière suivante :

```
# ipadm set-prop -p recv_buf=value tcp
```

Dans l'exemple suivant, la taille de la mémoire tampon est définie sur 164 KO.

```
# ipadm show-prop -p recv_buf tcp
PROTO PROPERTY   PERM CURRENT   PERSISTENT   DEFAULT  POSSIBLE
tcp   recv_buf   rw   128000       --        128000   2048-1048576

# ipadm set-prop -p recv_buf=164000 tcp
```

```
# ipadm show-prop -p recv_buf tcp
PROTO PROPERTY   PERM CURRENT   PERSISTENT   DEFAULT  POSSIBLE
tcp   recv_buf   rw   164000    164000       128000   2048-1048576
```

Il n'existe pas de valeur conseillée car elle varie selon la situation. Prenons les exemples suivants dans laquelle différentes valeurs sont définies pour les réseaux aux caractéristiques différentes BDP dans des conditions spécifiques :

1 Gbit/s standard avec une taille de tampon par défaut égale à 128 KO :	`BDP = 128 MBps * 0.001 s = 128 kB`
Théoriquement 1 Gbit /s WAN (Wide Area Network) avec latence de 100 ms :	`BDP = 128 MBps * 0.1 s = 12.8 MB`
Liaison Europe/ Etats-Unis (bande passante mesurée par la commande `uperf`)	`BDP = 2.6 MBps * 0.175 = 470 kB`

Si vous n'êtes pas en mesurer de déterminer le BDP, basez-vous sur les indications suivantes :

- Pour les transferts en bloc sur un réseau LAN, la valeur par défaut de la taille du tampon, 128 KO, suffit.
- Pour la plupart des déploiements WAN, la taille du tampon de réception doit être autour de 2 Mo.

Attention - L'augmentation de la taille de réception TCP accentue l'encombrement mémoire de nombreuses applications réseau.

L'affichage supplémentaires avec commande Statut du réseau `netstat`

You use the `netstat` commande pour générer l'affichage de l'état du réseau et des statistiques du protocole. Vous pouvez afficher le statut des UDP, TCP et SCTP points d'extrémité, sous forme de table, table de routage et d'afficher ainsi que les informations d'interface.

La commande `netstat` affiche plusieurs types de données du réseau selon l'option de la ligne de commande que vous spécifiez. Les informations affichées est particulièrement utile pour l'administration du système. Options utilisées pour les tâches fréquemment exécutées

sont décrites dans ce chapitre. Pour plus d'informations, reportez-vous à la page de manuel netstat(1M).

Cette section s'articule autour des rubriques suivantes :

Filtrage par type d'adresse de sortie netstat

Vous pouvez utiliser la commande netstat pour afficher l'état des réseaux IPv4 et IPv6. Vous pouvez choisir les informations de protocole à afficher en définissant la valeur de DEFAULT_IP dans le fichier /etc/default/inet_type ou en utilisant l'option -f dans la ligne de commande. En définissant la valeur de DEFAULT_IP, vous pouvez vous assurer que la commande netstat affiche uniquement les informations IPv4. Pour modifier ce paramètre, utilisez l'option -f de la ligne de commande. Pour plus d'informations sur le fichier inet_type, reportez-vous à la page de manuel inet_type(4).

Vous pouvez également utiliser l'option -fpour spécifier les arguments inet, inet6, unix (pour les sockets de domaine Unix utilisés pour les communications internes) ou sdp (Socket Description Protocol).

Displaying the Status of Sockets

Utilisez la commande netstat pour visualiser le statut des sockets sur un hôte local. Vous pouvez spécifier plusieurs options de commande netstat en tant qu'utilisateur standard.

Sockets afficher le statut de la commande connect comme suit :

```
% netstat
```

Affichez l'état des sockets, y compris les listener sockets non connectés, comme suit :

```
% netstat -a
```

EXEMPLE 1-3 L'affichage des sockets connectés

La sortie de la commande netstat contient de nombreuses statistiques. L'exemple ci-dessous illustre comment les sockets limiter la sortie à IPv4 uniquement.

```
% netstat -f inet

TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q    State
-------------------- -------------------- ----- ------ ----- ------ -----------
system-1.ssh          remote.38474     128872      0 128872      0 ESTABLISHED
system2.40721         remote.ldap      49232       0 128872      0 ESTABLISHED
```

Affichage des statistiques par protocole

L'option `netstat -s` permet d'afficher les statistiques des protocoles UDP, TCP, SCTP, ICMP et IP.

Affichage du statut des protocoles comme suit :

```
% netstat -s
```

Utilisez l'option `-P` pour filtrer la sortie de la commande `netstat` par protocole. Cette option n'est pas limitée à protocoles de transport.

Vous pouvez spécifier les valeurs suivantes avec cette option :

- `icmp`
- `icmpv6`
- `igmp`
- `ipv6tcp`
- `rawip`
- `sctp`
- `tcp`
- `udp`

Par exemple, vous pouvez filtrer la sortie `netstat` par le protocole UDP comme suit :

```
% netstat -s -P udp

UDP     udpInDatagrams      =  3704     udpInErrors         =     0
        udpOutDatagrams     =  3703     udpOutErrors        =     0
```

EXEMPLE 1-4 Affichage du statut des protocole TCP

L'exemple ci-dessous illustre comment afficher les résultats pour le protocole TCP en spécifiant l'option `-P`.

```
% netstat -P tcp

TCP: IPv4
Local Address     Remote Address    Swind Send-Q  Rwind Recv-Q      State
```

```
---------------- -------------------- ----- ------  ----- ------     -------
lhost-1.login      abc.def.local.Sun.COM.980 49640      0     49640    0 ESTABLISHED
lhost.login        ghi.jkl.local.Sun.COM.1020 49640     1     49640    0 ESTABLISHED
remhost.1014       mno.pqr.remote.Sun.COM.nfsd 49640    0     49640    0 TIME_WAIT

TCP: IPv6
Local Address    Remote Address        Swind Send-Q Rwind Recv-Q   State If
---------------- -------------------- ------ ----- ------ ----------- -----
localhost.38983   localhost.32777       49152      0 49152      0 ESTABLISHED
localhost.32777   localhost.38983       49152      0 49152      0 ESTABLISHED
localhost.38986   localhost.38980       49152      0 49152      0 ESTABLISHED
```

L'affichage du statut Interface réseau

L'option `i` de la commande `netstat` illustre le statut des interfaces réseau configurées sur le système local. Cette option permet de déterminer le nombre de paquets transmis et reçus sur un système sur les différents réseaux.

Afficher le statut des interfaces présents sur un réseau comme suit :

```
% netstat -i
```

L'exemple ci-dessous illustre comment utiliser la commande `netstat` avec une option de filtrage pour limiter l'impression d'une interface spécifique :

```
% netstat -i -I net0 -f inet
  Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
  net0  1500 abc.oracle.com abc.oracle.com 231001 0     55856  0     0      0
```

EXEMPLE 1-5 L'affichage du statut Interface réseau

L'exemple suivant illustre l'état du flux de paquets IPv4 et IPv6 sur les interfaces de l'hôte. Par exemple, le nombre de paquets entrants (`Ipkts`) affiché pour un serveur peut augmenter à chaque tentative de démarrage d'un client alors que le nombre de paquets sortants (`Opkts`) reste inchangé. Ce résultat suggère que le serveur détecte les paquets de demande d'initialisation envoyés par le client, Toutefois, le serveur ne parvient pas à formuler la réponse appropriée. Cette confusion peut être causée par une adresse incorrecte dans la base de données `hosts` ou `ethers`.

En revanche, si le nombre de paquets entrants reste inchangé sur la durée, l'ordinateur ne détecte même pas l'envoi des paquets. Ce résultat suggère un autre type d'erreur, vraisemblablement lié à un problème d'ordre matériel.

```
Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 loopback      localhost      142    0     142    0     0      0
net0  1500 host58        host58        1106302 0     52419  0     0      0

Name  Mtu  Net/Dest      Address                   Ipkts  Ierrs Opkts  Oerrs Collis
lo0   8252 localhost     localhost                  142    0     142    0     0
```

```
net0  1500 fe80::a00:20ff:feb9:4c54/10 fe80::a00:20ff:feb9:4c54 1106305 0 52422 0  0
```

L'affichage des informations sur les processus et de gestion des utilisateurs

La commande netstat inclut une nouvelle option -u. Cette option est sélectionnée, elle fournit des informations sur les utilisateurs et processus du système utilisant des ports spécifiques. Utilisez l'option u netstat, commande-pour afficher les ID utilisateur et le programme processus qui a créé le programme une extrémité de réseau ou le point d'extrémité qui contrôle actuellement le réseau.

Pour produire un meilleur alignement de la sortie netstat de la commande, vous pouvez spécifier l'option -n avec l'option -u. Pour plus d'informations et des exemples détaillés, reportez-vous à la page de manuel netstat(1M).

Affichage du statut des routes connues

L'option -r de la commande netstat permet d'afficher le tableau de routage de l'hôte local. Ce tableau affiche le statut de toutes les routes connues par un hôte.

```
% netstat -r
```

EXEMPLE 1-6 Sortie de table de routage obtenue à l'aide de la commande netstat

L'exemple suivant illustre le résultat de la commande netstat -r.

```
Routing Table: IPv4
Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
host15               myhost               U         1  31059  net0
10.0.0.14            myhost               U         1      0  net0
default              distantrouter        UG        1      2  net0
localhost            localhost            UH        42019361  lo0

Routing Table: IPv6
Destination/Mask      Gateway                           Flags Ref  Use   If
--------------------- --------------------------          ----- --- ------ -----
2002:0a00:3010:2::/64 2002:0a00:3010:2:1b2b:3c4c:5e6e:abcd U     1   0      net0:1
fe80::/10             fe80::1a2b:3c4d:5e6f:12a2            U     1   23     net0
ff00::/8              fe80::1a2b:3c4d:5e6f:12a2            U     1   0      net0
default               fe80::1a2b:3c4d:5e6f:12a2            UG    1   0      net0
localhost             localhost                            UH    9   21832  lo0
```

Le tableau suivant décrit les informations qui s'affiche dans la sortie de la commande netstat -r.

Paramètres	Description
Destination Destination/Mask	Spécifie l'hôte correspondant au point d'extrémité de destination de la route. Dans la table de routage IPv6, le point d'extrémité de destination est représenté par un préfixe de point d'extrémité de tunnel 6to4 (2002:0a00:3010:2::/64).
Gateway (Passerelle)	Spécifie la passerelle de transmission des paquets.
Flags (Indicateurs)	Indique le statut actuel de la route. L'indicateur U signifie que la route fonctionne. L'indicateur G signifie que la route mène à une passerelle.
Use	Affiche le nombre de paquets envoyés.
Interface	Indique l'interface de l'hôte local correspondant au point d'extrémité source de la transmission.

Affichage supplémentaire de l'état du réseau avec commande netstat.

Vous pouvez utiliser la commande netstat tout en permettant d'autres options de ligne de commande pour afficher d'états de réseau supplémentaires, en plus de ce qui est mentionné dans ce chapitre. Il y a 10 forme du produit et chaque table formulaire des statistiques relatives à un autre les différentes parties du sous-système de mise en réseau.

Certains des vous pouvez obtenir des statistiques supplémentaires, qui sont les suivantes :

netstat -g	Affiche les appartenances aux groupes de multidiffusion
netstat -P	Média affiche net-NDP ARP les mappings et les tables, procédez comme suit :
netstat -m	Affiche les statistiques de la mémoire STREAMS
netstat -M	Affiche le tableau de routage multidiffusion.
netstat -D	Les baux affiche le statut de DHCP
netstat -d	Tableau de cache de destination

Pour plus d'informations, reportez-vous à la page de manuel netstat(1M).

Exécution de l'administration TCP et UDP avec l'utilitaire netcat.

Utilisez l'utilitaire netcat (nc) pour réaliser une série de tâches qui sont associés à l'administration TCP ou UDP. Vous pouvez utiliser la commande pour les réseaux IPv4 et des réseaux IPv6.

Vous pouvez exécuter les tâches suivantes grâce à l'utilitaire `netcat` :

- Ouvrir les connexions TCP
- Envoyer les paquets UDP
- Et que le processus d'écoute reçoive les ports TCP UDP sur arbitraire
- Effectuer l'analyse de port

A la différence de la commande `telnet`, où chaque messages d'erreur est émis séparément pour une sortie standard, les messages d'erreur qui sont générés par les scripts `nc` sont combinés en une seule erreur standard, une méthode plus efficace.

L'utilitaire `netcat` prend en charge une nouvelle option `-M` qui vous permet de spécifier par socket du contrat les propriéts de niveau de service (SLA). Lorsque vous spécifiez les propriétés appropriées, le long un à l'aide de l'option `-M`, un flux MAC est créé pour le socket. Par exemple, vous pouvez utiliser l'option `-M` comme suit :

```
% nc -M maxbw=50M host.example.com 7777
% nc -l -M priority=high,inherit=on 2222
```

Comme décrit dans l'exemple précédent, l'option `-M` peut être utilisée pour spécifier une liste de propriétés SLA *name=value* séparées par des virgules.

Certaines méthodes d'installation n'installent pas le package logiciel `netcat` par défaut Vérifie si le package est installé sur votre système, comme suit :

```
% pkg list network/netcat
```

Si le package n'est pas installé, installez-le comme suit :

```
% pfexec pkg install pkg:/network/netcat
```

Pour de plus amples détails, reportez-vous à la page de manuel `netcat`(1).

Administration et journalisation des affichages de statut du réseau

Les tâches suivantes illustrent les procédures de vérification du statut du réseau à l'aide de commandes de réseau standard.

- “Contrôle de la sortie d'affichage des commandes IP” à la page 31
- “Actions de journalisation du démon de routage IPv4” à la page 32
- “Suivi des activités du démon de détection des voisins IPv6” à la page 32

Contrôle de la sortie d'affichage des commandes IP

Vous pouvez contrôler la sortie de la commande `netstat` pour afficher uniquement les informations IPv4, ou les informations IPv4 et IPv6.

1. **Créez le fichier `/etc/default/inet_type`.**

2. **Ajoutez l'une des entrées suivantes au fichier `/etc/default/inet_type` :**

   - **Pour afficher uniquement les informations IPv4, définissez la variable suivante :**

     ```
     DEFAULT_IP=IP_VERSION4
     ```

   - **Pour afficher les informations IPv4 et IPv6, définissez l'une des variables suivantes :**

     ```
     DEFAULT_IP=BOTH

     DEFAULT_IP=IP_VERSION6
     ```

Pour plus d'informations, reportez-vous à la page de manuel `inet_type`(4).

Remarque - L'indicateur `-f` dans la commande `netstat` remplace les valeurs dans le fichier `inet_type`.

Exemple 1-7 Contrôle de la sortie à afficher des informations IPv4 et IPv6

L'exemple suivant présente la sortie lorsque vous spécifiez la variable DEFAULT_IP = BOTH ou la variable DEFAULT_IP = IP_VERSION6 dans le fichier : `inet_type`

```
# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
net0: flags=100001004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4,PHYSRUNNING> mtu 1500 index
 603
        inet 10.46.86.54 netmask ffffff00 broadcast 10.46.8.255
        ether 0:1e:68:49:98:80
lo0: flags=2002000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
        inet6 ::1/128
net0: flags=120002000841<UP,RUNNING,MULTICAST,IPv6,PHYSRUNNING> mtu 1500 index 603
        inet6 fe80::21e:68ff:fe49:9880/10
        ether 0:1e:68:49:98:80
net0:1: flags=120002080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6,PHYSRUNNING> mtu 1500 index 603
        inet6 2001:b400:414:6059:21e:68ff:fe49:9880/64
```

Si vous spécifiez la variable `DEFAULT_IP=IP_VERSION4` dans le fichier `inet_type`, la sortie suivante s'affiche :

```
# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
net0: flags=100001004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4,PHYSRUNNING> mtu 1500 index
 603
        inet 10.46.86.54 netmask ffffff00 broadcast 10.46.8.255
        ether 0:1e:68:49:98:80
```

Actions de journalisation du démon de routage IPv4

Lorsque vous soupçonnez un dysfonctionnement du démon de routage IPv4 `routed`, vous pouvez créer un journal permettant d'effectuer le suivi de l'activité correspondante. Le journal inclut tous les transferts de paquets à compter du démarrage du démon `routed`.

Créer un fichier journal des actions de effectue le suivi du démon de routage comme suit :

```
# /usr/sbin/in.routed /var/log-file-name
```

Attention - Sur les réseaux à forte activité, la sortie de cette commande peut être générée sur une base quasi continue.

L'exemple suivant illustre le début du journal créé lorsque vous effectuez la procédure. “Actions de journalisation du démon de routage IPv4” à la page 32.

```
-- 2003/11/18 16:47:00.000000 --
Tracing actions started
RCVBUF=61440
Add interface lo0  #1   127.0.0.1      -->127.0.0.1/32
<UP|LOOPBACK|RUNNING|MULTICAST|IPv4> <PASSIVE>
Add interface net0 #2   10.10.48.112    -->10.10.48.0/25
<UP|BROADCAST|RUNNING|MULTICAST|IPv4>
turn on RIP
Add    10.0.0.0        -->10.10.48.112      metric=0  net0  <NET_SYN>
Add    10.10.48.85/25  -->10.10.48.112      metric=0  net0  <IF|NOPROP>
```

▼ Suivi des activités du démon de détection des voisins IPv6

Si vous pensez que le démon IPv6 `in.ndpd` ne fonctionne pas correctement, vous pouvez générer le suivi de l'activité correspondante. Le suivi s'affiche sur la sortie standard jusqu'à

l'arrêt du processus. Il inclut tous les transferts de paquets à compter du démarrage du démon in.ndpd.

1. **Générez le suivi du démon `in.ndpd`.**

   ```
   # /usr/lib/inet/in.ndpd -t
   ```

2. **Pour arrêter le processus de suivi, appuyez sur les touches Ctrl-C.**

Exemple 1-8 La fonction de suivi du démon in.ndpd

La sortie suivante illustre le début du suivi du démon in.ndpd.

```
# /usr/lib/inet/in.ndpd -t
Nov 18 17:27:28 Sending solicitation to  ff02::2 (16 bytes) on net0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:b9:4c:54>
Nov 18 17:27:28 Received valid advert from fe80::a00:20ff:fee9:2d27 (88 bytes) on net0
Nov 18 17:27:28         Max hop limit: 0
Nov 18 17:27:28         Managed address configuration: Not set
Nov 18 17:27:28         Other configuration flag: Not set
Nov 18 17:27:28         Router lifetime: 1800
Nov 18 17:27:28         Reachable timer: 0
Nov 18 17:27:28         Reachable retrans timer: 0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:e9:2d:27>
Nov 18 17:27:28         Prefix: 2001:08db:3c4d:1::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
Nov 18 17:27:28         Prefix: 2002:0a00:3010:2::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
```

Sondage des hôtes distants à l'aide de la commande `ping`

Vous pouvez utiliser la commande ping pour déterminer si votre système peut communiquer avec un hôte distant. Lors de l'exécution de la commande ping, le protocole ICMP envoie un datagramme à l'hôte spécifié et attend la réponse. Le protocole ICMP permet de gérer les erreurs se produisant sur les réseaux TCP/IP. Lorsque vous utilisez la commande ping, vous pouvez déterminer si les paquets IP pourront être communiqués entre votre hôte et un hôte distant spécifié.

La syntaxe suivante est la syntaxe de base de la commande ping :

/usr/sbin/ping *host* **[***timeout***]**

où *host* est le nom de l'hôte distant. L'argument *timeout* indique la durée en secondes pendant laquelle la commande ping tente de contacter l'hôte distant. La valeur par défaut est de 20 secondes. Pour plus d'informations, reportez-vous à la page de manuel ping(1M).

Modification de la commande ping en vue de la prise en charge IPv6

La commande ping se sert des protocoles IPv4 et IPv6 pour sonder les hôtes cibles. Le choix du protocole dépend des adresses renvoyées par le serveur de noms pour l'hôte cible spécifique. Par défaut, si ce serveur renvoie une adresse IPv6 pour l'hôte cible, la commande ping utilise le protocole IPv6. S'il renvoie une adresse IPv4, la commande ping utilise le protocole IPv4. Pour ignorer cette action, vous pouvez taper l'option -A dans la ligne de commande et spécifier le protocole à utiliser.

Pour plus d'informations, reportez-vous à la page de manuel ping(1M).

La vérification permettant de déterminer si une vérification de l'accessibilité de l'hôte distant

Pour déterminer si un hôte distant peut être atteint; utilisez la commande ping comme suit :

```
% ping hostname
```

Si l'hôte processus d'écoute accepte les transmissions ICMP, le message suivant s'affiche :

```
hostname is alive
```

Ce message indique que l'hôte a répondu à la requête ICMP. Toutefois, si l'hôte ne peut pas recevoir l'est vers le bas les paquets ICMP, ou s'il y en a une gamme problème entre la configuration de l'hôte et l'hôte distant, génère la réponse suivante depuis la commande ping :

```
no answer from hostname
```

Détermination de l'existence de paquets rejetés entre votre hôte et un hôte distant

La perte de paquets peut rendre les connexions réseau lentes, parce qu'il faut du temps en plus pour retransmettre les données rejetées. Utilisez l'option -s de la commande ping afin de savoir si des paquets sont rejetés entre votre hôte et un hôte distant :

```
% ping -s hostname
```

La commande `ping -s`*nom-hôte* envoie des paquets en continu à l'hôte spécifié pendant un laps de temps donné ou jusqu'à l'envoi d'un caractère d'interruption.

```
% ping -s host1.domain8
PING host1.domain8 : 56 data bytes
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=0. time=1.67 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=1. time=1.02 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=2. time=0.986 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=3. time=0.921 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=4. time=1.16 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.00 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.980 ms

^C

----host1.domain8  PING Statistics----
7 packets transmitted, 7 packets received, 0% packet loss
round-trip (ms)  min/avg/max/stddev = 0.921/1.11/1.67/0.26
```

La statistique de perte de paquets indique si l'hôte a des paquets rejetés. Si la commande `ping` indique une perte de paquets, contrôlez l'état du réeau à l'aide des commandes `ipadm` et `netstat`. Reportez-vous à " Contrôle d'interfaces et d'adresses IP " du manuel " Configuration et administration des composants réseau dans Oracle Solaris 11.2 " et "L'affichage supplémentaires avec commande Statut du réseau `netstat`" à la page 24 pour de plus amples détails.

Affichage des informations de routage à l'aide de la commande `traceroute`

La commande `traceroute` permet d'obtenir le suivi de la route empruntée par un paquet IP pour accéder à un système distant. La commande `traceroute` permet de détecter les erreurs de configuration de routage et les échecs de chemin de routage. Si un hôte est inaccessible, la commande `traceroute` permet d'afficher le chemin suivi par les paquets afin de détecter les emplacements susceptibles d'être à l'origine de l'échec.

La commande `traceroute` affiche également le délai d'aller-retour de chaque passerelle sur le chemin d'accès à l'hôte cible. Ces informations permettent notamment de déterminer l'emplacement des ralentissements de trafic entre les deux hôtes.

Pour plus de détails sur la commande `traceroute` command, reportez-vous à la page de manuel `traceroute`(1M).

Cette section s'articule autour des rubriques suivantes :

- "Modification de la commande `traceroute` en vue de la prise en charge IPv6" à la page 36

Modification de la commande `traceroute` en vue de la prise en charge IPv6

Vous pouvez exécuter la commande `traceroute` pour tracer les routes IPv4 et IPv6 vers un hôte spécifique. Du point de vue du protocole, la commande `traceroute` utilise le même algorithme que la commande `ping`. Pour ignorer ce choix, tapez l'option `-A` dans la ligne de commande. Vous pouvez tracer chaque route vers chaque adresse d'un hôte à multihébergement en tapant l'option `-a` dans la ligne de commande.

Localisation de la route menant à un hôte distant

Utilisez la commande `traceroute` comme suit pour déterminer la route menant à un système distant :

```
% traceroute destination-hostname
```

La sortie suivante de la commande `traceroute` affiche le chemin à sept sauts suivi par les paquets pour circuler du système local `nearhost` vers le système distant `farhost`. La sortie illustre également le temps nécessaire à un paquet pour traverser les différents sauts.

```
% traceroute farhost
traceroute to farhost (172.16.64.39), 30 hops max, 40 byte packets
1  frbldg7c-86 (172.16.86.1)  1.516 ms  1.283 ms  1.362 ms
2  bldg1a-001 (172.16.1.211)  2.277 ms  1.773 ms  2.186 ms
3  bldg4-bldg1 (172.16.4.42)  1.978 ms  1.986 ms  13.996 ms
4  bldg6-bldg4 (172.16.4.49)  2.655 ms  3.042 ms  2.344 ms
5  ferbldg11a-001 (172.16.1.236)  2.636 ms  3.432 ms  3.830 ms
6  frbldg12b-153 (172.16.153.72)  3.452 ms  3.146 ms  2.962 ms
7  farhost (172.16.64.39)  3.430 ms  3.312 ms  3.451 ms
```

Suivi de toutes les routes

Utilisez la commande `traceroute` l'option `-a` pour le suivi de toutes les routes sur un système local :

```
% traceroute -a host-name
```

L'exemple suivant affiche toutes les routes possibles pour accéder à un hôte double pile :

```
% traceroute -a v6host
```

```
traceroute: Warning: Multiple interfaces found; using 2001:db8:4a3a:1:56:a0:a8 @ net0:2
 traceroute to v6host (2001:db8:4a3b:5:102:a00:fe79:19b0),30 hops max, 60 byte packets
1 v6-rout86 (2001:db8:4a3b:1:56:a00:fe1f:59a1) 35.534 ms 56.998 ms *
2 2001:db8::255:0:c0a8:717 32.659 ms 39.444 ms *
3  farhost (2001:db8:4a3b:2:103:a00:fe9a:ce7b)  401.518 ms  7.143 ms *
4  distant (2001:db8:4a3b:3:100:a00:fe7c:cf35)  113.034 ms  7.949 ms *
5  v6host (2001:db8:4a3b:5:102:a00:fe79:19b0)  66.111 ms *  36.965 ms *

traceroute to v6host  (192.168.10.75),30 hops max,40 byte packets
1  v6-rout86 (172.16.86.1)  4.360 ms  3.452 ms  3.479 ms
2  flrmpj17u (172.16.17.131)  4.062 ms  3.848 ms  3.505 ms
3  farhost (10.0.0.23)  4.773 ms *  4.294 ms
4  distant (192.168.10.104)  5.128 ms  5.362 ms *
5  v6host  (192.168.15.85)  7.298 ms  5.444 ms *
```

L'analyse et Network Wireshark du trafic à l'aide Analysers tshark

Tshark est un analyseur du trafic réseau basé sur une ligne de commande qui permet de capturer des paquets de données d'un réseau opérationnel ou de lire des paquets d'un *fichier de capture* enregistré en imprimant une forme décodée de ces paquets dans le formulaire la sortie standard ou en écrivant les paquets dans un fichier Sans aucune option, la commande Tshark fonctionne de la même façon que la commande `tcpdump` et utilise également le même format de capture de fichier `libpcap`. En outre, la détection tshark, la lecture est capable de la même capture, ainsi que l'écriture de fichiers en tant que Wireshark ceux qui sont pris en charge par.

Wireshark est une interface utilisateur graphique (GUI) de tiers d'un analyseur de protocole réseau utilisé de pour vider et analyser le trafic du réseau de façon interactive. Similaire à la commande `snoop`, vous pouvez utiliser Wireshark pour parcourir des données de paquet sur un réseau opérationnel ou d'un fichier de capture enregistré auparavant. Par défaut, Wireshark utilise le format `libpcap` pour des captures de fichier, également utilisé par l'utilitaire `tcpdump` et les autres outils similaires. Un avantage clé de l'utilisation est la capacité de lire et d'importer plusieurs autres formats de fichiers, en dehors du format `libpcap`.

Mise à disponibilité de plusieurs commandes tshark et à la fois des fonctions uniques, Wireshark, y compris les éléments suivants :

- Tous les capable d'élaboration TCP de la conversation paquets au sein d'un et d'afficher les données, de cette conversation hexadécimal dans format EBCDIC ou ASCII
- Champs contiennent une quantité filtrables des analyseurs et à d'autres de protocole réseau
- Utilisent une syntaxe des analyseurs plus riche pour que d'autres valeurs de création de filtres de protocole réseau

Pour utiliser sur votre tshark Oracle Solaris et système Wireshark, vérifiez d'abord que l'installation des packages logiciels et, si nécessaire, installez-les comme suit :

```
# pkg install tshark
```

```
# pkg install wireshark
```

Pour plus d'informations, reportez-vous aux pages de manuel tshark(1) et wireshark(1).

Reportez-vous également à la documentation Wireshark à l'adresse http://www.wireshark.org/.

Contrôle du transfert des paquets à l'aide de la commande snoop

Vous pouvez utiliser la commande snoop pour contrôler le trafic de réseau. La commande snoop permet de capturer les paquets réseau et l'affichage du contenu dans le format spécifié. Les paquets peuvent être affichés dès leur réception ou dès l'enregistrement dans un fichier. L'écriture des données dans un fichier intermédiaire par la commande snoop permet de réduire la probabilité de perte de paquet liée à l'activité de suivi. Le fichier est alors également interprété par la commande snoop.

Pour capturer des paquets en provenance et à destination de l'interface par défaut en mode promiscuité, vous devez prendre le rôle d'administrateur réseau ou de root. Dans sa forme contractée, la commande snoop affiche uniquement les données en rapport avec le protocole principal. Par exemple, un paquet NFS affiche uniquement les informations NFS. Les informations RPC, UDP, IP et Ethernet sont supprimées, mais vous pouvez y accéder en sélectionnant l'une des options détaillées de la commande.

L'exécution répétée à intervalles fréquents de la commande snoop permet d'identifier les comportements normaux du système. Pour plus de détails à propos de la commande snoop, reportez-vous à la page de manuel snoop(1M).

Cette section s'articule autour des rubriques suivantes :

Modification de la commande snoop en vue de la prise en charge IPv6

La commande snoop peut capturer des paquets IPv4 et IPv6 . Cette commande peut s'afficher avec des en-têtes IPv6, des en-têtes d'extension IPv6, des en-têtes ICMPv6 et des données

de protocole Neighbor Discovery. Par défaut, la commande `snoop` affiche les deux types de paquet (IPv4 et IPv6). Pour afficher soit le mot clé du protocole `ip` ou `ip6`, la commande affiche uniquement des paquets IPv4 ou IPv6. L'option de filtrage IPv6 vous permet de filtrer tous les paquets IPv4 et IPv6 et d'afficher uniquement les paquets IPv6. Reportez-vous à “Contrôle du trafic réseau IPv6” à la page 41 et à la page de manuel `snoop`(1M).

▼ Vérification des paquets en provenance de toutes les interfaces

1. **Imprimez les informations sur les interfaces connectées au système.**

   ```
   # ipadm show-if
   ```

   La commande `snoop` utilise normalement le premier périphérique non-loopback, en principe, l'interface réseau principale.

2. **Commencez la capture des paquets en exécutant `snoop` sans arguments.**

   ```
   # snoop
   ```

3. **Pour arrêter le processus, appuyez sur les touches Ctrl-C.**

Exemple 1-9 L'affichage de base de sortie `snoop`

L'exemple suivant affiche les valeurs de sortie de la commande `snoop` pour un hôte double pile.

```
# snoop
Using device /dev/net (promiscuous mode)
router5.local.com -> router5.local.com ARP R 10.0.0.13, router5.local.com is
0:10:7b:31:37:80
router5.local.com -> BROADCAST     TFTP Read "network-confg" (octet)
myhost -> DNSserver.local.com      DNS C 192.168.10.10.in-addr.arpa. Internet PTR ?
DNSserver.local.com  myhost        DNS R 192.168.10.10.in-addr.arpa. Internet PTR
niserve2.
.
.
.
fe80::a00:20ff:febb:e09 -> ff02::9 RIPng R (5 destinations)
```

Dans la sortie précédente, les paquets capturés montrent une requête et une réponse DNS, ainsi que des paquets ARP (Address Resolution Protocol) en provenance du routeur local et des publications de l'adresse IPv6 lien-local sur le démon `in.ripngd`.

Vérification des paquets à partir d'un groupe IPMP

Dans Oracle Solaris 10, si vous souhaitez capturer des paquets à partir d'un groupe IPMP, vous devrez exécuter manuellement la `snoop` sur chacun des interfaces du groupe IPMP. Dans la version 11 d'Oracle Solaris, vous pouvez capturer tous les paquets de toutes les interfaces d'un groupe IPMP à l'aide de la commande `snoop` avec l'option `-I`. La sortie est ensuite consolidée en un seul flux de sortie.

La commande `snoop` utilise normalement le premier périphérique non-loopback, en principe, l'interface réseau principale. En revanche, lorsque l'option `-I` est utilisée, la commande `snoop` utilise les interfaces IP (à partir de `/dev/ipnet/*`) comme affichées par la commande `ipadm show-if`. Vous pouvez également utiliser cette option pour activer le trafic loopback et d'autres constructions IP. L'option `-d` utilise les interfaces de liaison de données qui apparaissent dans la sortie de la commande `dladm show-link`.

1. **Imprimez les informations sur les interfaces connectées au système.**

   ```
   # ipadm show-if
   ```

2. **Indiqué commencez la capture des paquets IPMP pour le groupe.**

   ```
   # snoop -I ipmp-group
   ```

3. **Pour arrêter le processus, appuyez sur les touches Ctrl-C.**

Capture de la sortie de la commande `snoop` dans un fichier

1. **Capturez une session de commande `snoop` dans un fichier. Ainsi,**

   ```
   # snoop -o /tmp/cap
   Using device /dev/eri (promiscuous mode)
   30 snoop: 30 packets captured
   ```

   Dans cet exemple, 30 paquets sont capturés dans le fichier `/tmp/cap`. Ce fichier peut se trouver dans tout répertoire contenant suffisamment d'espace disque. Le nombre de paquets capturés s'affiche sur la ligne de commande. Vous pouvez dès lors appuyez sur les touches Ctrl-C à tout moment pour arrêter le processus.

   La commande `snoop` génère une charge réseau conséquente, ce qui risque de fausser légèrement les résultats. Pour garantir la précision des résultats, exécutez la commande `snoop` à partir d'un système tiers.

2. **Consultez le fichier de capture de sortie de la commande `snoop`.**

```
# snoop -i filename
```

Exemple 1-10 L'affichage de capture de sortie `snoop`

La sortie suivante illustre diverses captures susceptibles d'être obtenues suite à l'exécution de `snoop -i`.

```
# snoop -i /tmp/cap
1   0.00000 fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375
ICMPv6 Neighbor advertisement
...
10  0.91493    10.0.0.40 -> (broadcast)  ARP C Who is 10.0.0.40, 10.0.0.40 ?
34  0.43690 nearserver.here.com  -> 224.0.1.1  IP  D=224.0.1.1 S=10.0.0.40 LEN=28,
ID=47453, TO =0x0, TTL=1
35  0.00034  10.0.0.40 -> 224.0.1.1    IP  D=224.0.1.1 S=10.0.0.40 LEN=28, ID=57376,
TOS=0x0, TTL=47
```

▼ Vérification des paquets transmis entre un client et un serveur IPv4

1. **Définissez un système snoop `un` partir d'un hub port (ou d'un port mis en miroir sur un commutateur) qui est connecté à la base de données du client ou du serveur.**

   Le système tiers (système `snoop`) vérifie tous les types de trafic entre les deux ordinateurs. Le suivi obtenu grâce à la commande `snoop` reflète donc le transfert réel de données.

2. **Exécutez la commande `snoop`associée aux options appropriées, puis enregistrez la sortie dans un fichier.**

3. **Consultez et interprétez la sortie.**

   Reportez-vous au document RFC 1761, Snoop Version 2 Packet Capture File Format (http://www.rfc-editor.org/rfc/rfc1761.txt) pour plus d'informations sur le fichier de capture `snoop`.

Contrôle du trafic réseau IPv6

Utilisez la commande `snoop` de la manière suivante pour capturer des paquets IPv6 :

```
# snoop ip6
```

L'exemple suivant illustre la sortie standard qui pourraient s'afficher lorsque vous exécutez la commande `snoop ip6` sur un noeud.

```
# snoop ip6
```

```
fe80::a00:20ff:fecd:4374 -> ff02::1:ffe9:2d27 ICMPv6 Neighbor solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
solicitation
fe80::a00:20ff:febb:e09 -> ff02::9      RIPng R (11 destinations)
fe80::a00:20ff:fee9:2d27 -> ff02::1:ffcd:4375 ICMPv6 Neighbor solicitation
```

Pour plus d'informations sur la commande `snoop`, reportez-vous à la page de manuel snoop(1M).

Contrôle des paquets à l'aide de périphériques de couche IP

Les périphériques de couche IP ont été introduits dans Oracle Solaris pour améliorer l'observabilité. Ces périphériques donnent accès à tous les paquets avec les adresses associées à l'interface réseau du système. Ces adresses incluent des adresses locales ainsi que des adresses hébergées sur des interfaces sans loopback ou des interfaces logiques. Le trafic observable peut correspondre aux adresses IPv4 et IPv6. Par conséquent, vous pouvez surveiller l'ensemble du trafic destiné au système. Le trafic peut être du trafic d'IP avec loopback, des paquets provenant de machines distantes, des paquets envoyés à partir du système ou la totalité du trafic transféré.

Les périphériques de couche IP permettent à l'administrateur d'une zone globale de surveiller le trafic entre les zones ainsi qu'au sein d'une zone. L'administrateur d'une zone non globale peut également observer le trafic envoyé et reçu par cette zone.

Pour surveiller le trafic sur la couche IP, utilisez la commande `snoop` avec l'option `-I`, plus récente. Cette option indique à la commande d'utiliser les nouveaux périphériques de couche IP plutôt que le périphérique sous-jacent de couche liaison pour afficher les données de trafic.

▼ Vérification des paquets sur la couche IP

1. **(Facultatif) Si nécessaire, imprimez les informations sur les interfaces connectées au système.**

   ```
   # ipadm show-if
   ```

2. **Capturez le trafic IP sur une interface spécifique.**

   ```
   # snoop -I interface [-V | -v]
   ```

Paquets méthodes for Checking

Tous les exemples suivants sont basés sur cette configuration système, procédez comme suit :

```
# ipadm show-addr

ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           dhcp     ok           10.153.123.225/24
lo0/v6            static   ok           ::1/128
net0/v6           addrconf ok           fe80::214:4fff:2731:b1a9/10
net0/v6           addrconf ok           2001:0db8:212:60bb:214:4fff:2731:b1a9/64
net0/v6           addrconf ok           2001:0db8:56::214:4fff:2731:b1a9/64
```

Supposons que deux zones, `sandbox` et `toybox`, utilisent les adresses IP suivantes :

- `sandbox` – `172.0.0.3`
- `toybox` – `172.0.0.1`

Vous pouvez exécuter la commande `snoop -I` sur les différentes interfaces du système. L'affichage des informations du paquet dépend de si vous êtes administrateur de la zone globale ou de la zone non globale.

EXEMPLE 1-11 Trafic sur l'interface loopback

L'exemple suivant illustre le résultat de la commande `snoop`pour l'interface loopback.

```
# snoop -I lo0
Using device ipnet/lo0 (promiscuous mode)
localhost -> localhost    ICMP Echo request (ID: 5550 Sequence number: 0)
localhost -> localhost    ICMP Echo reply (ID: 5550 Sequence number: 0)
```

Pour générer une sortie détaillée, utilisez l'option `-v`.

```
# snoop -v -I lo0
Using device ipnet/lo0 (promiscuous mode)
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 1 arrived at 10:40:33.68506
IPNET:  Packet size = 108 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 4
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 0
IPNET:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
...
```

La prise en charge de l'observation des paquets sur la couche IP introduit un nouvel en-tête `ipnet` qui précède les paquets observés. Les ID de source et de destination sont tous deux indiqués. L'ID `0` indique que le trafic est généré à partir de la zone globale.

EXEMPLE 1-12 L'observation de paquets réalisée à l'aide de l'option de flux périphérique `net0` dans les zones locales.

L'exemple suivant montre le trafic des différentes zones qui se trouvent dans le système. Vous pouvez voir tous les paquets associés aux adresses IP `net0`, y compris les paquets livrés localement aux autres zones. Si vous générez une sortie détaillée, vous pouvez voir les zones impliquées dans le flux de paquets.

```
# snoop -I net0
Using device ipnet/net0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=62117 Syn Seq=195630514 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=62117 S=22 Syn Ack=195630515 Seq=195794440 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=62117 Ack=195794441 Seq=195630515 Len=0 Win=49152
sandbox -> toybox TCP D=62117 S=22 Push Ack=195630515 Seq=195794441 Len=20 Win=491
```

```
# snoop -I net0 -v port 22
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 5 arrived at 15:16:50.85262
IPNET:  Packet size = 64 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 0
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 1
IPNET:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:         .... ..0. = not ECN capable transport
IP:         .... ...0 = no ECN congestion experienced
IP:   Total length = 40 bytes
IP:   Identification = 22629
IP:   Flags = 0x4
IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 6 (TCP)
IP:   Header checksum = 0000
IP:   Source address = 172.0.0.1, 172.0.0.1
IP:   Destination address = 172.0.0.3, 172.0.0.3
IP:   No options
IP:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 46919
TCP:  Destination port = 22
TCP:  Sequence number = 3295338550
```

```
TCP:  Acknowledgement number = 3295417957
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        0... .... = No ECN congestion window reduced
TCP:        .0.. .... = No ECN echo
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP         .... 0... = No push
TCP         .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 49152
TCP:  Checksum = 0x0014
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
```

Dans la sortie précédente, l'en-tête `ipnet` indique que le paquet provient de la zone globale (ID `0`) to `sandbox` (ID `1`).

EXEMPLE 1-13 Un réseau du trafic par identification de l'observation Zone

L'exemple suivant montre la manière d'observer le trafic réseau en identifiant la zone qui est extrêmement utile pour les systèmes dotés de plusieurs zones. Actuellement, la zone s'identifie uniquement par l'intermédiaire de l'ID de zone. L'utilisation de la commande `snoop` avec les noms de zone n'est pas prise en charge.

```
# snoop -I hme0 sandboxsnoop -I net0 sandbox
Using device ipnet/hme0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=61658 Syn Seq=374055417 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=61658 S=22 Syn Ack=374055418 Seq=374124525 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=61658 Ack=374124526 Seq=374055418 Len=0 Win=49152
```

L'observation du trafic avec les commandes réseau `ipstattcpstat`.

Deux nouvelles commandes pour l'observation de deux types de trafic réseau sur un serveur sont ajoutées à cette version : `ipstat` et `tcpstat`.

La commande`ipstat` sert à collecter et générer des rapports statistiques sur le trafic IP sur un serveur en fonction du mode de sortie sélectionné et un ordre de tri qui est spécifié dans la syntaxe de commande. Cette commande vous permet d'observer le trafic réseau à ce sur la source IP, couche de destination, ces valeurs étant agrégées, protocole de couche supérieure et de l'interface. Utilisez cette commande lorsque vous souhaitez observer le trafic entre un serveur et d'autres serveurs.

La commande tcpstat permet de rassembler et d'établir des rapports sur les statistiques sur le trafic UDP TCP sur un serveur en fonction du mode de sortie sélectionné et un ordre de tri qui est spécifié dans la syntaxe de commande. Cette commande vous permet d'observer le trafic réseau à la couche de transport TCP et UDP, plus particulièrement en. En plus des adresses IP source et de destination, vous pouvez consulter ou les applications source et cible, le TCP PID ports UDP du processus à l'origine le trafic est, l'envoi ou de la réception et le nom de la zone dans laquelle cet processus est en cours d'exécution.

Voici une sélection des manières dont vous utilisez la commande : tcpstat

- Identifier les plus grandes le trafic TCP et UDP sources sur un serveur.
- Examiner le trafic qui est généré par un processus donné.
- Examiner le trafic qui est en cours de génération depuis une zone particulière.
- Déterminer le traitement est liée à un port local.

Remarque - La liste n'est pas exhaustive. Il existe plusieurs manières d'utiliser la command tcpstat. Reportez-vous à la page de manuel tcpstat(1M) pour plus d'informations.

Pour utiliser les commandes, ipstatà tcpstat, l'un des privilèges suivants est requis :

- Prenez le rôle root
- Le privilège dtrace_kernel doit être octroyé de façon explicite.
- Etre affectés en tant qu'administrateur réseau profil de droits ou Network Observability

Les exemples suivants montrent différentes méthodes s'offrent à vous pouvez utiliser ces deux commandes pour observer le trafic réseau. Pour plus d'informations, reportez-vous aux pages de manueltcpstat(1M) et ipstat(1M)

L'exemple suivant illustre la sortie de la commande ipstat exécutée avec l'option -c. Utilisez l'option-c pour imprimer les rapports plus récents après des rapports antérieurs si vous n'écrasez pas le rapport précédent. Le chiffre 3 dans cet exemple montre comment spécifier l'intervalle d'affichage des données, qui est le même que si la commande ipstat 3 avait été appelée.

```
# ipstat -c 3
SOURCE                    DEST                      PROTO    INT       BYTES
zucchini                  antares                   TCP      net0      72.0
zucchini                  antares                   SCTP     net0      64.0
antares                   zucchini                  SCTP     net0      56.0
amadeus.foo.example.com   10.6.54.255               UDP      net0      40.0
antares                   zucchini                  TCP      net0      40.0
zucchini                  antares                   UDP      net0      16.0
antares                   zucchini                  UDP      net0      16.0
Total: bytes in: 192.0  bytes out: 112.0
```

Par comparaison, l'exemple suivant illustre la sortie de la commande tcpstat utilisée avec l'option -c :

```
# tcpstat -c 3
```

```
ZONE         PID PROTO  SADDR             SPORT DADDR             DPORT   BYTES
global    100680 UDP    antares           62763 agamemnon          1023   76.0
global    100680 UDP    antares             775 agamemnon          1023   38.0
global    100680 UDP    antares             776 agamemnon          1023   37.0
global    100680 UDP    agamemnon          1023 antares           62763   26.0
global    104289 UDP    zucchini          48655 antares            6767   16.0
global    104289 UDP    clytemnestra      51823 antares            6767   16.0
global    104289 UDP    antares            6767 zucchini         48655   16.0
global    104289 UDP    antares            6767 clytemnestra     51823   16.0
global    100680 UDP    agamemnon          1023 antares             776   13.0
global    100680 UDP    agamemnon          1023 antares             775   13.0
global    104288 TCP    zucchini          33547 antares            6868    8.0
global    104288 TCP    clytemnestra      49601 antares            6868    8.0
global    104288 TCP    antares            6868 zucchini         33547    8.0
global    104288 TCP    antares            6868 clytemnestra     49601    8.0
Total: bytes in: 101.0  bytes out: 200.0
```

Les exemples supplémentaires suivants montrent encore d'autres méthodes pour observer le trafic de votre réseau moyennant les commandes `ipstat` et `tcpstat`.

EXEMPLE 1-14 Observation des cinq flux de trafic les plus actifs moyennant la commande `ipstat`.

L'exemple suivant fait état IP les cinq la plus active les flux de trafic. L'option-`l` *nlines* spécifie le nombre de lignes de données pour la sortie par rapport.

```
# ipstat -l 5
SOURCE                     DEST                       PROTO    IFNAME     BYTES
charybdis.foo.example.com  achilles.exampl            UDP      net0        6.6K
eratosthenes.example.com   aeneas.example.c           TCP      tun0        6.1K
achilles.exampl            charybdis.foo.example.com  UDP      net0       964.0
aeneas.example.c           eratosthenes.example.com   TCP      tun0       563.0
odysseus.example.          255.255.255.255            UDP      net0        66.0
Total: bytes in: 12.6K bytes out:  2.2K
```

EXEMPLE 1-15 Affichage d'un horodateur à l'aide de la commande `ipstat`

L'exemple suivant répertorie le principal trafic IP moyennant l'horodatage au format de date standard (`-d` d). Vous pouvez indiquer que l'horodatage soit imprimé en secondes ou en heure Unix (`-d` u). L'intervalle est défini sur 10 secondes.

```
# ipstat -d d -c 10
Monday, March 26, 2012 08:34:07 PM EDT
SOURCE                     DEST                       PROTO    IFNAME     BYTES
charybdis.foo.example.com  achilles.exampl            UDP      net0       15.1K
eratosthenes.example.com   aeneas.example.c           TCP      tun0       13.9K
achilles.exampl            charybdis.foo.example.com  UDP      net0        2.4K
aeneas.example.c           eratosthenes.example.com   TCP      tun0        1.5K
odysseus.example.          255.255.255.255            UDP      net0       66.0
cassiopeia.foo.example.com aeneas.example.c           TCP      tun0       29.0
aeneas.example.c           cassiopeia.foo.example.com TCP      tun0       20.0
Total: bytes in: 29.1K bytes out:  3.8K
```

EXEMPLE 1-16 Observation des cinq flux de trafic les plus actifs à l'aide de la commande tcpstat.

L'exemple suivant fait état TCP les cinq des flux de trafic la plus active un serveur, procédez comme suit :

```
# tcpstat -l 5
ZONE            PID PROTO  SADDR            SPORT DADDR            DPORT   BYTES
global        28919 TCP    achilles.exampl  65398 aristotle.exampl   443   33.0
zone1          6940 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
zone1          6940 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
global         8350 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
global         8350 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
Total: bytes in: 16.0  bytes out: 49.0
```

EXEMPLE 1-17 Affichage des informations d'horodatage moyennant la commande tcpstat

Dans l'exemple suivant, la commande tcpstat est utilisée pour pour afficher les informations d'horodatage au trafic réseau sur un serveur TCP au format de date standard :

```
# tcpstat -d d -c 10
Saturday, March 31, 2012 07:48:05 AM EDT
ZONE            PID PROTO  SADDR            SPORT DADDR            DPORT   BYTES
global         2372 TCP    penelope.example 58094 polyphemus.examp    80   37.0
zone1          6940 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
zone1          6940 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
global         8350 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
global         8350 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
Total: bytes in: 16.0  bytes out: 53.0
```

♦ ♦ ♦ CHAPITRE 2

IPMP d'administration sur

La fonctionnalité de chemins d'accès multiples sur réseau IP (IPMP) est une technologie de niveau 3 qui vous permet de regrouper plusieurs interfaces IP en une seule et même interface logique. Grâce à ses fonctions de détection de défaillance, basculement d'accès transparent et répartition de la charge des paquets, IPMP améliore les performances du réseau en assurant une disponibilité constante au système.

Ce chapitre aborde les sujets suivants :

Remarque - Tout au long de ce chapitre et dans le Chapitre 3, Administration d'IPMP, toutes les références au terme *interface* signifient, plus spécifiquement, *interface IP*. A moins qu'un processus de qualification n'indique explicitement uns autre utilisation du terme, telle que carte réseau (NIC), le terme se réfère toujours à l'interface qui est configurée sur la couche IP.

Nouveautés dans IPMP

Les fonctions suivantes ont été ajoutées ou modifiées dans cette version.

Les modifications de la configuration IPMP

Dans cette version le multipathing sur réseau IP fonctionne différemment que dans Oracle Solaris 10. Un changement important est que les interfaces IP sont dorénavant regroupées en une interface IP *virtuelle* (`ipmp0` par exemple). L'interface IP virtuelle sert toutes les adresses IP de données, tandis que les adresses de test utilisées pour la détection de défaillance basée sur sonde sont assignées à une interface sous-jacente telle que `net0`. Pour plus d'informations,

reportez-vous à"Fonctionnement de la fonctionnalité de chemins d'accès multiples sur réseau IP (IPMP)" à la page 56.

Reportez-vous au workflow général suivant lors de la transition de votre configuration IPMP existante vers le nouveau modèle IPMP :

1. Assurez - vous que le profil `DefaultFixed` est activé sur votre système avant de configurer IPMP. Reportez-vous à " Activation et désactivation des profils " du manuel " Configuration et administration des composants réseau dans Oracle Solaris 11.2 "
2. Assurez-vous que les adresses MAC sur les systèmes SPARC sont uniques. Reportez-vous à " Garantie de l'unicité de l'adresse MAC de chaque interface " du manuel " Configuration et administration des composants réseau dans Oracle Solaris 11.2 "
3. Exécutez la commande `dladm` pour configurer les liaisons de données. Pour utiliser les mêmes périphériques de réseau physique au sein de votre configuration IPMP, vous devez d'abord identifier les liaisons de données associées à chaque instance de périphérique :

```
# dladm show-phys
LINK              MEDIA               STATE      SPEED  DUPLEX    DEVICE
net1              Ethernet            unknown    0      unknown   bge1
net0              Ethernet            up         1000   full      bge0
net2              Ethernet            unknown    1000   full      e1000g0
net3              Ethernet            unknown    1000   full      e1000g1
```

4. Si vous avez déjà utilisé `e1000g0` et `e1000g1` pour votre configuration, vous devez désormais utiliser `net2` et `net3`. Notez que les liaisons de données ne peuvent pas être uniquement basées sur les liens physiques mais aussi sur les groupements, VLAN, VNIC, et ainsi de suite. Pour plus d'informations, reportez-vous à" Affichage des liaisons de données d'un système " du manuel " Configuration et administration des composants réseau dans Oracle Solaris 11.2 ".
5. Utilisez la commande `ipadm` pour effectuer les tâches suivantes :
   - Configurer la couche réseau.
   - Créer des interfaces IP.
   - Ajouter des interfaces IP au groupe IPMP.
   - Ajouter des adresses de données à un groupe IPMP.

Pour plus d'informations sur les modifications de la commande d'administration réseau dans cette version, reportez-vous à " Modifications apportées à la commande d'administration réseau " du manuel " Transition d'Oracle Solaris 10 vers Oracle Solaris 11.2 ".

Prise en charge d'IPMP dans Oracle Solaris

IPMP fournit la prise en charge suivante :

- IPMP vous permet de configurer plusieurs interfaces IP dans un seul groupe connu sous le nom de groupe IPMP. Dans son ensemble, le groupe IPMP comptant plusieurs interfaces IP

sous-jacentes est représenté sous la forme d'une seule *interface IPMP*. Cette interface est considérée comme n'importe quelle autre interface de la couche IP de la pile réseau. Toutes les tâches administrative IP, tables de routage, tables protocole de résolution d'adresse (ARP), règles de pare-feu et autres procédures liées aux IP fonctionnent avec un groupe IPMP faisant référence à l'interface IPMP.

- Le système gère la distribution d'aresses de données parmi les interfaces actives sous-jacentes. Lorsque le groupe IPMP est créé, les adresses de données appartiennent à l'interface IPMP sous forme de *address pool*. Le noyau lie alors automatiquement et aléatoirement les adresses de données aux interfaces actives sous-jacentes du groupe.
- Vous utilisez la commande `ipmpstat` principalement pour obtenir des informations sur les groupes IPMP. Cette commande fournit des informations sur tous les aspects de la configuration IPMP, tels que les interfaces IP sous-jacentes du groupe, les adresses de données et de test, les types de détection de défaillance en cours d'utilisation et les interfaces défaillantes. Les fonctions de commande `ipmpstat`, les options que vous pouvez utiliser et les sorties que chaque option génère sont décrites dans “Surveillance des informations IPMP” à la page 91.
- Vous pouvez affecter une interface IPMP un nom personnalisé pour identifier le groupe IPMP plus facilement. Reportez-vous à la section “Configuration de groupes IPMP” à la page 71.

Avantages de l'utilisation de la fonctionnalité de chemins d'accès multiples sur réseau IP

Différents facteurs peuvent expliquer le fait qu'une interface devient inutilisable, notamment une défaillance ou la mise hors ligne à des fins de maintenance. Sans IPMP, le système ne peut plus être contacté par le biais d'une des adresses IP associées à cette interface inutilisable. En outre, les connexions existantes qui utilisent ces adresses IP sont interrompues.

Grâce à IPMP, vous pouvez configurer plusieurs interfaces IP dans un *groupe IPMP*. Le groupe fonctionne comme une interface IP avec des adresses de données pour envoyer ou recevoir le trafic réseau. Si une interface sous-jacente dans le groupe échoue, les adresses de données sont réparties entre les autres interfaces actives sous-jacentes du groupe. Ainsi, le groupe permet de maintenir la connectivité réseau malgré un échec de l'interface. Avec IPMP, la connectivité réseau est toujours disponible, à condition qu'au moins une interface soit utilisable pour le groupe.

En outre, IPMP répartit le trafic réseau sortant sur l'ensemble des interfaces du groupe IPMP, ce qui permet d'améliorer les performances réseau globales. Ce processus est appelé *répartition de charge sortante*. De plus, le système contrôle indirectement la répartition de charge entrante en effectuant une sélection des adresses source pour les paquets dont l'adresse IP source n'a pas été spécifiée par l'application. Cependant, si une application a explicitement choisi une adresse IP source, le système ne remplace pas cette adresse.

Prenez en compte les points suivants IPMP les informations relatives aux stratégies pour les messages entrants et qui applique la répartition de charge sortante, procédez comme suit :

- Pour les adresses IP on-link, IPMP sélectionne au hasard une seule interface d'IP active pour atteindre cette adresse IP. Dans le cas où il existerait plusieurs connexions distinctes pour une même adresse IP on-link, toutes ces connexions utiliseront la même interface IP sortante. En outre, si les modifications d'interface IP sur une échelle de temps, la modification a une incidence sur tous les connexions à ce IP de l'adresse.
- Pour les adresses IP hors liaison, IPMP sélectionne au hasard une seule interface IP pour atteindre l'adresse IP du routeur IP on-link via lequel l'adresse IP hors liaison sera atteinte. Cette stratégie signifie en pratique que IP que toutes les adresses hors liaison IPMP pour un groupe d'utiliser IP donné interface IP unique.

Remarque - Entrantes en cours et la répartition de charge sortante les stratégies sont susceptibles d'être modifiées.

Les groupements de liaisons IPMP exécutent des fonctions similaires à IPMP pour améliorer les performances et la disponibilité du réseau. Pour comparer ces deux technologies, reportez-vous àAnnexe A, “ Groupement de liaisons et IPMP : comparaison des fonctionnalités ” du manuel “ Gestion des liaisons de données réseau dans Oracle Solaris 11.2 ”.

Règles d'utilisation de la fonctionnalité de chemins d'accès multiples sur réseau IP

La configuration du groupe IPMP est déterminée par votre configuration système spécifique.

IPMP observez les règles suivantes pour la configuration, procédez comme suit :

1. Il faut configurer plusieurs interfaces IP sur le même réseau local (LAN) dans un groupe IPMP. Le terme LAN fait généralement référence à une grande variété de configurations réseau locales, notamment aux réseaux locaux virtuels (VLAN) et aux réseaux locaux filaires ou sans fil dont les noeuds appartiennent au même domaine de diffusion au niveau de la couche de liaison.

Remarque - Si plusieurs groupes IPMP appartiennent au même domaine de diffusion (L2), ces groupes ne sont pas pris en charge. Un domaine de diffusion L2 est généralement mappé à un sous-réseau spécifique. Par conséquent, vous ne devez configurer qu'un seul groupe IPMP par sous-réseau. Notez également que certaines exceptions à cette règle s'appliquent, par exemple, dans le cas de certains systèmes techniques qui sont fournis par Oracle. Pour obtenir de plus de clarté, contactez votre représentant du support technique Oracle.

---

2. Les interfaces IP sous-jacentes d'un groupe IPMP ne doivent pas s'étendre sur plusieurs réseaux locaux.

   Par exemple, supposons qu'un système avec trois interfaces est connecté à deux LAN séparés. Deux interfaces IP sont connectées à un même LAN, tandis que la dernière est connectée à l'autre LAN. Dans ce cas, les deux interfaces IP se connectant au premier LAN doivent être configurées en tant que groupe IPMP, comme requis par la première règle. En conformité avec la deuxième règle, l'interface IP unique qui se connecte au second LAN ne peut pas devenir membre de ce groupe IPMP. Aucune configuration IPMP n'est requise pour l'interface IP unique. Cependant, vous pouvez configurer l'interface unique au sein d'un groupe IPMP pour surveiller la disponibilité de l'interface. La configuration IPMP à interface unique est abordée plus en détail dans la section “Types de configurations d'interface IPMP” à la page 55.

   Prenons un autre cas où la liaison au premier LAN se compose de trois interfaces IP alors que l'autre se compose de deux interfaces. Pour cette installation, il faut configurer deux groupes IPMP : un groupe de trois interfaces connecté au premier LAN et un groupe de deux interfaces connecté au second.

3. Toutes les interfaces d'un même groupe doivent disposer de modules STREAMS configurés selon le même ordre. Dans la planification d'un groupe IPMP, vous devez d'abord vérifier l'ordre des modules STREAMS sur toutes les interfaces du futur groupe IPMP, puis empilez les modules de chaque interface en respectant l'ordre standard pour le groupe IPMP. Vous pouvez imprimer une liste de modules STREAMS à l'aide de la commande `ifconfig` *interface* `modlist`. Voici un exemple de sortie de la commande `ifconfig` pour une interface `net0` :

   ```
   # ifconfig net0 modlist
   0 arp
   1 ip
   2 e1000g
   ```

   Comme le montre la sortie, les interfaces existent normalement en tant que pilotes réseau directement sous le module IP. Aucune configuration supplémentaire n'est nécessaire sur ces interfaces. Cependant, certaines technologies sont empilées sous la forme d'un module STREAMS entre le module IP et le pilote de réseau. Dans le cas d'un module STREAMS avec conservation de statut, vous pouvez observer des comportements inattendus lors d'un basculement, même si vous empilez le même module sur toutes les interfaces d'un groupe. Cependant, vous pouvez utiliser des modules STREAMS sans état, à condition que vous les déplaciez selon le même ordre sur toutes les interfaces du groupe IPMP.

   L'exemple suivant illustre la commande pouvant être utilisées pour utiliser pour empilez les modules de chaque interface en IPMP respectant l'ordre standard pour le groupe, procédez comme suit :

   ```
   # ifconfig net0 modinsert vpnmod@3
   ```

   Pour obtenir des instructions détaillées sur la planification d'un groupe IPMP, reportez-vous à “Planification pour un groupe IPMP” à la page 72.

Composants IPMP

Les points suivants sont des composants logiciels d'IPMP :

- **Démon de chemins d'accès multiples** (`in.mpathd`) - Détecte les défaillances et et réparations de l'interface. Le démon procède à la détection de défaillance basée sur les liaisons et la détection de défaillance basée sur les sondes si les adresses de test sont configurées pour les interfaces sous-jacentes. Selon le type de méthode de détection de défaillance suivie, le démon définit ou supprime les indicateurs appropriés sur l'interface afin de signaler que l'interface a subi une défaillance ou a été réparée. Le démon peut également être configuré pour surveiller la disponibilité de toutes les interfaces, y compris de celles qui n'appartiennent pas à un groupe IPMP. Pour une description de la détection de défaillance, reportez-vous à la section “Détection de défaillance dans IPMP” à la page 63.

  Le démon `in.mpathd` contrôle également la désignation des interfaces actives dans le groupe IPMP. Le démon tente de conserver le même nombre d'interfaces initialement configuré lorsque le groupe IPMP a été créé. Par conséquent, `in.mpathd` active ou désactive les interfaces sous-jacentes selon les besoins pour être cohérent avec la stratégie configurée de l'administrateur. Pour plus d'informations sur la manière dont le démon `in.mpathd` gère l'activation des interfaces sous-jacentes, reportez-vous à la section “Fonctionnement de la fonctionnalité de chemins d'accès multiples sur réseau IP (IPMP)” à la page 56. Pour plus d'informations sur le démon, reportez-vous à la page de manuel `in.mpathd`(1M).
- Le **module de noyau IP** gère la répartition de la charge sortante en associant l'ensemble des adresses IP disponibles dans le groupe à l'intégralité des interfaces IP sous-jacentes disponibles dans le groupe. Ce module effectue également une sélection d'adresses source pour gérer la répartition de la charge entrante. Les deux rôles du module améliorent les performances en termes de trafic réseau.
- **fichier de configuration IPMP** (`/etc/default/mpathd`) - Définit le comportement du démon.

  Il convient de personnaliser ce fichier pour définir les paramètres suivants :

  - Interfaces cible à tester lors de l'exécution de la détection de défaillance basée sur sonde
  - Durée du test d'une cible en vue de détecter des défaillances
  - Statut à attribuer à une interface défaillante après sa réparation
  - Etendue des interfaces IP à surveiller (précisant s'il faut également inclure interfaces IP du système qui ne font pas partie de groupes IPMP)

  Pour plus d'informations sur les procédures de modification du fichier de configuration, reportez-vous à la “Configuration du comportement du démon IPMP ” à la page 89.
- La `commande` **ipmpstat** offre différents types d'informations sur le statut global du multipathing sur réseau IP. Cette commande affiche également d'autres informations spécifiques sur les interfaces IP sous-jacentes pour chaque groupe, ainsi que les adresses de données et de test configurées pour le groupe. Pour plus d'informations sur cette commande,

reportez-vous à “Surveillance des informations IPMP” à la page 91 et à la page de manuel `ipmpstat`(1M).

Types de configurations d'interface IPMP

En règle générale, une configuration IPMP se compose d'au moins deux interfaces physiques situées sur le même système et connectées au même LAN.

Ces interfaces peuvent appartenir à un groupe IPMP dans l'une des configurations suivantes :

- **Configuration active-active** : groupe IPMP dans lequel toute les interfaces sous-jacentes sont actives. Un *interface active* est une interface IP qui est actuellement disponible pour l'utilisation par le groupe IPMP.

Remarque - Par défaut, une interface sous-jacente devient active lorsque vous la configurez pour faire partie d'un groupe IPMP.

- **Configuration active-de réserve** : groupe IPMP qui comprend au moins une *interface de réserve*. Bien qu'inactive, l'interface de secours est surveillée par le démon de fonctionnalité de chemins d'accès multiples pour effectuer le suivi de la disponibilité de l'interface, en fonction de sa configuration. Si la notification de défaillance de liaison est prise en charge par l'interface, la détection de défaillance basée sur les liaisons est utilisée. Si l'interface est configurée avec une adresse de test, la détection de défaillance basée sur sonde est également utilisée. Si une interface active échoue, l'interface de secours est déployée automatiquement en fonction des besoins. Vous pouvez configurer autant d'interfaces de réserve que vous le souhaitez pour un groupe IPMP.

Vous pouvez également configurer une interface unique IPMP est possible de le faire dans son propre groupe. Un groupe IPMP à une seule interface se comporte de la même façon qu'un groupe IPMP avec plusieurs interfaces. Cependant, cette configuration IPMP ne fournit pas de haute disponibilité pour le trafic réseau. Si l'interface sous-jacente échoue, le système perd toute capacité à envoyer ou recevoir du trafic. La configuration d'un groupe IPMP à une interface présente l'intérêt de surveiller la disponibilité de l'interface par le biais de la détection de défaillance. Après la définition d'une adresse de test sur l'interface, le démon de fonctionnalité de chemins d'accès multiples peut effectuer le suivi de l'interface à l'aide de la détection de défaillance basée sur sonde.

En règle générale, une configuration de groupe IPMP à une interface vient compléter d'autres technologies bénéficiant de plus grandes capacités de basculement, comme le logiciel Oracle Solaris Cluster. Le système peut continuer à surveiller l'état de l'interface sous-jacente, mais le logiciel Oracle Solaris Cluster fournit les fonctionnalités permettant de garantir la disponibilité du réseau en cas de panne. Pour plus d'informations sur le logiciel Oracle Solaris Cluster, reportez-vous au manuel “ Oracle Solaris Cluster Concepts Guide ”.

Un groupe IPMP sans interfaces sous-jacente peut également exister, tel qu'un groupe dont les interfaces ont été supprimées. Le groupe IPMP n'est pas détruit, mais il ne peut pas être utilisé pour envoyer et recevoir du trafic. Les interfaces sous-jacentes sont mises en ligne pour le groupe, puis les adresses de données de l'interface IPMP sont allouées à ces interfaces et le système reprend le traitement du trafic réseau.

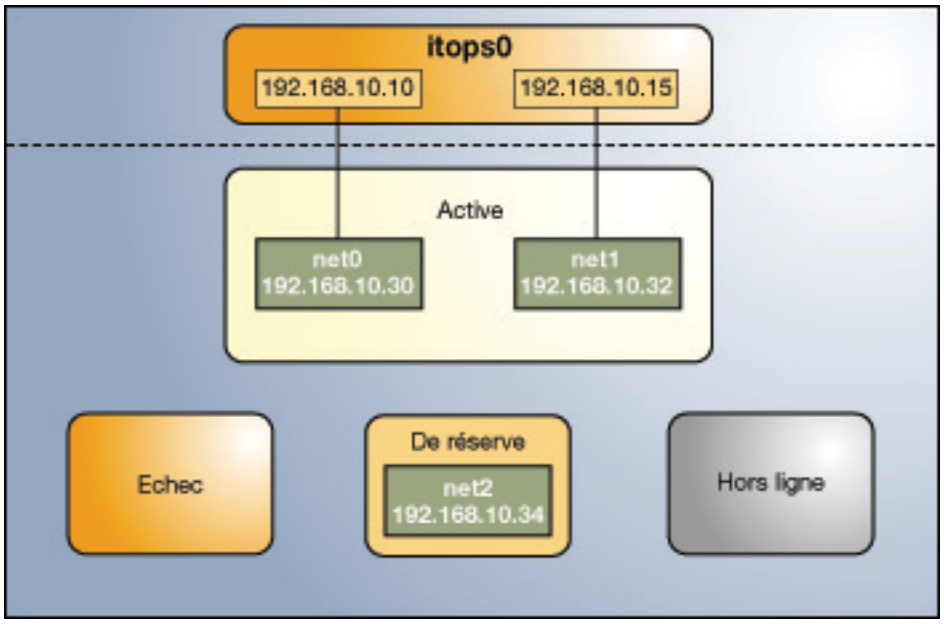
Fonctionnement de la fonctionnalité de chemins d'accès multiples sur réseau IP (IPMP)

IPMP assure la disponibilité du réseau en tentant de préserver le nombre d'interfaces actives et de secours initialement configurées à la création du groupe IPMP.

La détection de défaillance IPMP peut être basée sur les liaisons ou les sondes (ou les deux) pour déterminer la disponibilité d'une l'interface IP sous-jacente spécifique du groupe. Si IPMP détermine qu'une interface sous-jacente a échoué, alors cette interface est marquée comme ayant échoué et n'est plus utilisable. L'adresse IP de données qui a été associée à l'interface défaillante est ensuite redistribuées à une autre interface opérationnelle du groupe. Si elle est disponible, une interface de secours est également déployée pour maintenir le nombre initial d'interfaces actives.

Considérez un groupe IPMP à trois interfaces, `itops0`, avec une configuration active de secours, comme le montre la figure suivante.

FIGURE 2-1 Configuration IPMP active-de secours



Le groupe IPMP `itops0` est configuré comme suit :

- Deux adresses de données sont affectées au groupe : `192.168.10.10` et `192.168.10.15`.
- Deux interfaces sous-jacentes sont configurées en tant qu'interfaces actives et des noms de liaison flexibles leur sont affectés : `net0` et `net1`.
- Le groupe possède une interface de secours, également avec un nom de liaison flexible : `net2`.
- La détection de défaillance par sonde étant utilisée, les interfaces actives et de secours sont configurées avec les adresses de test suivantes :
  - `net0`: `192.168.10.30`
  - `net1`: `192.168.10.32`
  - `net2`: `192.168.10.34`

Remarque - Les zones Active, Hors ligne, De secours et Echec de Figure 2-1, “Configuration IPMP active-de secours”, Figure 2-2, “Panne d'interface dans IPMP ”, Figure 2-3, “Panne d'interface de secours dans IPMP ” et Figure 2-4, “Processus de récupération IPMP” présentent uniquement l'état des interfaces sous-jacentes, et non leur emplacement physique. Aucun mouvement physique d'interfaces ou d'adresses, ni transfert d'interfaces IP ne se produit au sein de cette implémentation IPMP. Les zones servent uniquement à montrer comment une interface sous-jacente change d'état suite à une panne ou une réparation.

Vous pouvez exécuter la commande `ipmpstat` avec différentes options pour afficher des types d'informations spécifiques à propos de groupes IPMP existants. Pour d'autres exemples, reportez-vous à la section “Surveillance des informations IPMP” à la page 91.

La commande ci-dessous affiche des informations sur la configuration IPMP de la Figure 2-1, “Configuration IPMP active-de secours” :

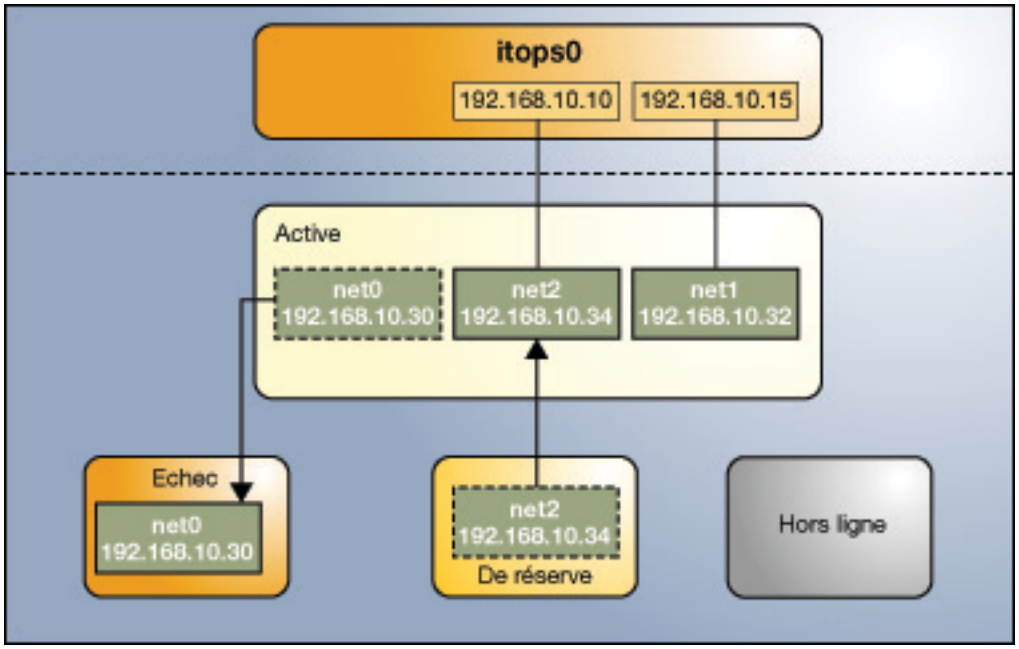
```
# ipmpstat -g
GROUP     GROUPNAME     STATE     FDT        INTERFACES
itops0    itops0        ok        10.00s     net1 net0 (net2)
```

Vous devez afficher des informations sur les interfaces sous-jacentes du groupe comme suit :

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS      LINK        PROBE     STATE
net0        yes        itops0    -------    up          ok        ok
net1        yes        itops0    --mb---    up          ok        ok
net2        no         itops0    is-----    up          ok        ok
```

IPMP permet de maintenir la disponibilité du réseau en gérant les interfaces sous-jacentes afin de conserver le nombre initial d'interfaces actives. Par conséquent, si `net0` subit une défaillance, `net2` est déployée pour garantir que le groupe IPMP compte toujours deux interfaces actives. L'activation `net2` est illustrée ci-dessous.

FIGURE 2-2 Panne d'interface dans IPMP



Remarque - Le mappage univoque d'adresses de données sur des interfaces actives dans la Figure 2-2, “Panne d'interface dans IPMP ” ne sert qu'à simplifier l'illustration. Le module de noyau IP peut en effet attribuer des adresses de données de façon aléatoire sans nécessairement adhérer à une relation univoque entre des adresses de données et des interfaces.

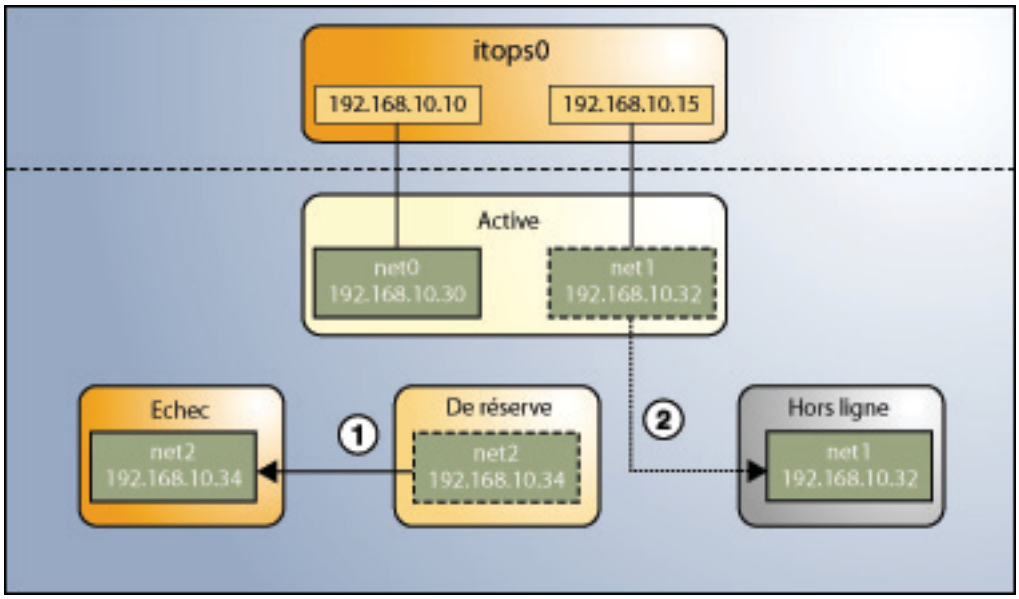
La commande `ipmpstat` affiche les informations de la figure comme suit :

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS      LINK        PROBE     STATE
net0        no         itops0    -------    up          failed    failed
net1        yes        itops0    --mb---    up          ok        ok
net2        yes        itops0    -s-----    up          ok        ok
```

Une fois réparée, `net0` revient à son état d'interface active. A son tour, `net2` revient à son état initial d'attente.

La Figure 2-3, “Panne d'interface de secours dans IPMP ” illustre un autre scénario de panne dans lequel l'interface de secours `net2` subit une défaillance (1). Par la suite, l'administrateur met hors ligne l'interface active `net1` (2). Le résultat est que le groupe IPMP est laissé avec une seule interface opérationnelle, `net0`.

FIGURE 2-3 Panne d'interface de secours dans IPMP

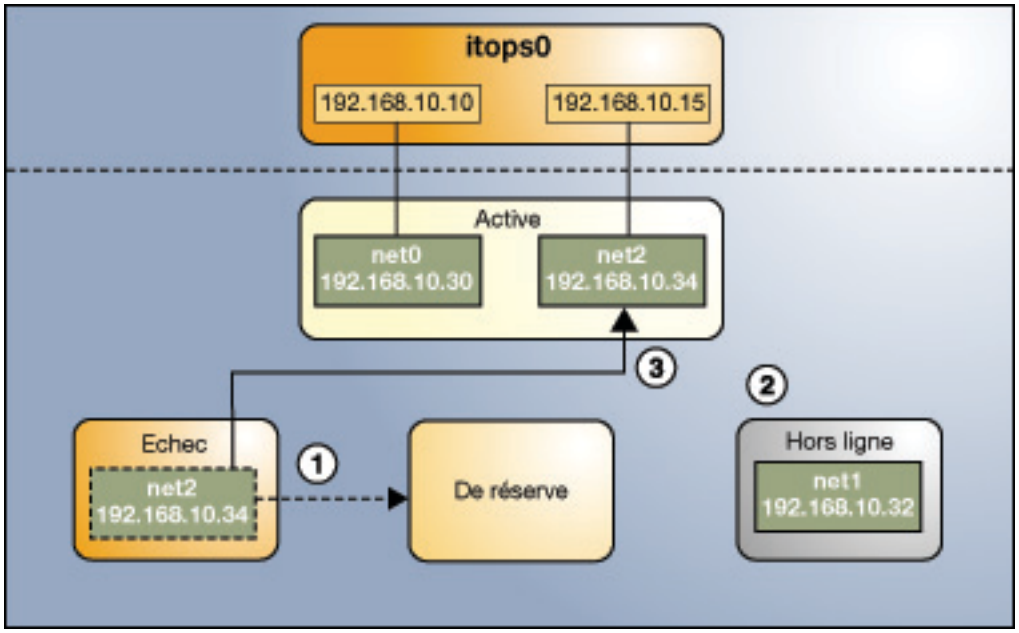


La commande `ipmpstat` affiche les informations de la figure comme suit :

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS       LINK        PROBE     STATE
net0        yes        itops0    -------     up          ok        ok
net1        no         itops0    --mb-d-     up          ok        offline
net2        no         itops0    is-----     up          failed    failed
```

Pour cette panne, la récupération après réparation d'une interface se comporte différemment. Le processus de récupération dépend du nombre initial d'interfaces actives dans le groupe IPMP par rapport à la configuration après réparation. La figure suivante représente le processus de récupération.

FIGURE 2-4 Processus de récupération IPMP



Dans la Figure 2-4, “Processus de récupération IPMP”, lorsque `net2` est réparée, elle doit normalement reprendre son état d'origine d'interface de secours (1). Cependant, le groupe IPMP ne reflète toujours pas le nombre initial de deux interfaces actives, car `net1` est toujours hors ligne (2). Par conséquent, IPMP déploie `net2` comme interface active (3).

La commande `ipmpstat` affiche le scénario IPMP post-réparation comme suit :

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS       LINK        PROBE     STATE
net0        yes        itops0    -------     up          ok        ok
net1        no         itops0    --mb-d-     up          ok        offline
net2        yes        itops0    -s-----     up          ok        ok
```

Un processus de récupération similaire est mis en oeuvre si l'incident concerne une interface active également configurée en mode `FAILBACK=no`, dans lequel une interface active en panne ne revient pas automatiquement à l'état actif après réparation. Supposons que `net0` dans la Figure 2-2, “Panne d'interface dans IPMP ” soit configurée en mode `FAILBACK=no`. Dans ce mode, `net0` devient une interface de secours après sa réparation, même s'il s'agissait initialement d'une interface active. L'interface `net2` demeure active afin de conserver le nombre initial de deux interfaces actives dans le groupe IPMP.

La commande `ipmpstat` affiche les informations de récupération comme suit :

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS      LINK        PROBE     STATE
net0        no         itops0    i------    up          ok        ok
net1        yes        itops0    --mb---    up          ok        ok
net2        yes        itops0    -s-----    up          ok        ok
```

Pour plus d'informations sur ce type de configuration, reportez-vous à la section “Mode `FAILBACK=no`” à la page 67.

Adressage IPMP

Vous pouvez configurer la détection de défaillance IPMP sur des réseaux IPv4, ainsi que sur des réseaux IPv4 et IPv6 double pile. Les interfaces configurées avec IPMP prennent deux types d'adresses en charge : les adresses de données et les adresses test. Les adresses IP résident sur le groupe IPMP (le groupe) *uniquement* et sont indiqués comme adresses de données. Les adresses IP d'adresses de test qui résident sur les interfaces sous-jacentes .

Adresses de données

Les *adresses de données* désignent les adresses IPv4 et IPv6 conventionnelles attribuées à une interface IP de manière dynamique au moment de l'initialisation par le serveur DHCP ou manuellement par le biais de la commande `ipadm`. Les adresses de données sont affectées au groupe IPMP interface (ou groupe) *uniquement*. Le trafic de paquets standard IPv4 et, le cas échéant, IPv6, est considéré comme du *trafic de données*. Le trafic de données utilise les adresses de données hébergées sur l'interface IPMP et le flux à travers les interfaces actives de ce groupe ou cette interface IPMP.

Adresses de test

Les adresses de test sont des adresses IPMP sèécifiques utilisées par le démon `in.mpathd` pour exécuter une détection de défaillance et réparation basée sur sonde. Des adresses de test peuvent également être affectées de façon dynamique par le serveur DHCP, ou manuellement avec la commande `ipadm`. Seules des adresses de test sont associées aux interfaces sous-jacentes du groupe IPMP. Lorsqu'une interface sous-jacente échoue, l'adresse de test de l'interface continue d'être utilisé par le démon `in.mpathd` pour la détection de défaillance basée sur sonde pour vérifier la réparation qui suit.

Remarque - Il faut configurer des adresses de test uniquement si vous souhaitez activer la détection de défaillance basée sur sonde. Dans le cas contraire, vous pouvez activer le test transitif pour détecter la défaillance sans utiliser des adresses de test. Pour de plus amples informations sur la détection de défaillance basée sur sonde avec ou sans adresses de test, reportez-vous à “Détection de défaillance basée sur sonde” à la page 64.

Dans les précédentes implémentations IPMP, il fallait marquer les adresses de test comme `DEPRECATED` pour empêcher leur utilisation par les applications, en particulier en cas de défaillance d'interface. Dans l'implémentation actuelle, des adresses de test se trouvent dans les interfaces sous-jacentes. Ainsi, ces adresses ne peuvent plus être utilisées accidentellement par des applications qui ne sont pas compatibles avec IPMP. Cependant, pour vous assurer que ces adresses ne sont pas considérées comme une source possible pour les paquets de données, le système les marque automatiquement avec l'indicateur `NOFAILOVER` ainsi que `DEPRECATED`.

Vous pouvez définir l'adresse IPv4 de votre choix sur le sous-réseau en tant qu'adresse de test. Dans la mesure où les adresses IPv4 sont une ressource limitée pour de nombreux sites, il est préférable dans certains cas d'utiliser des adresses privées RFC 1918 non acheminables en tant qu'adresses test. Notez que le démon `in.mpathd` n'échange que des sondes ICMP avec d'autres hôtes situés sur le même sous-réseau que l'adresse test. Si vous utilisez des adresses test de type RFC 1918, veillez à configurer d'autres systèmes, des routeurs de préférence, sur le réseau avec des adresses situées sur le sous-réseau RFC 1918. Le démon `in.mpathd` peut ensuite échanger les sondes avec des systèmes cible. Pour plus d'informations sur les adresses privées RFC 1918, reportez-vous au document RFC 1918, Address Allocation for Private Internets (http://www.rfc-editor.org/rfc/rfc1918.txt) (en anglais).

La seule adresse de test IPv6 valide correspond à l'adresse link-local d'une interface physique. Inutile de définir une adresse IPv6 distincte en tant qu'adresse de test IPMP. L'adresse link-local IPv6 est basée sur l'adresse MAC (Media Access Control) de l'interface. Les adresses link-local sont configurées automatiquement lorsque l'interface devient compatible IPv6 au moment de l'initialisation ou lorsque l'interface est configurée manuellement par le biais de la commande `ipadm`.

Lorsqu'IPv4 et IPv6 sont montés sur la totalité des interfaces d'un groupe IPMP, il est inutile de configurer des adresses test IPv4 distinctes. Le démon `in.mpathd` peut utiliser des adresses link-local IPv6 en tant qu'adresses test.

Détection de défaillance dans IPMP

Afin de garantir une disponibilité continue du réseau pour envoyer ou recevoir le trafic, IPMP effectue une détection de défaillance sur les interfaces d'IP sous-jacentes du groupe IPMP. Les interfaces défaillantes restent inutilisables jusqu'à ce qu'elles soient réparées. Les interfaces actives restantes continuent de fonctionner et les interfaces de secours sont déployées en fonction des besoins.

Le démon `in.mpathd` gére les types de détection de défaillance suivants :

- Deux types de détection de défaillance basée sur sonde, procédez comme suit :
  - Aucune adresse de test n'est configurée (test transitif).
  - Des adresses de test sont configurées.
- Détection de défaillance basée sur les liaisons si la prise en charge est assurée par le pilote NIC

Détection de défaillance basée sur sonde

La détection de défaillance basée sur sonde consiste à utiliser des sondes ICMP pour vérifier si une interface a échoué. L'implémentation de cette méthode de détection de défaillance dépend de l'utilisation d'adresses de test.

Détection de défaillance basée sur sonde avec des adresses de test

Cette méthode de détection de défaillance repose sur l'envoi et la réception de messages de sonde ICMP utilisant des adresses de test. Ces messages (également appelés *trafic de sonde* ou *trafic de test*) sont envoyés de l'interface à un ou plusieurs systèmes cible sur le même réseau local. Le démon `in.mpathd` examine toutes les cibles séparément à travers toutes les interfaces qui ont été configurées pour la détection de défaillance basée sur sonde. Si, après cinq tests consécutifs sur une interface donnée, aucune réponse n'est obtenue, `in.mpathd` considère que l'interface est défaillante. La fréquence des tests dépend du *temps de détection de défaillance* (*FDT*). Le temps de détection de défaillance par défaut est de 10 secondes. Cependant, vous pouvez régler le temps de détection de défaillance dans le fichier de configuration IPMP. Pour obtenir des instructions, reportez-vous à “Configuration du comportement du démon IPMP ” à la page 89.

Pour optimiser la détection de défaillance basée sur sonde, il faut définir plusieurs systèmes cible pour recevoir les sondes du démon `in.mpathd`. En présence de plusieurs systèmes cibles, vous pouvez mieux déterminer la nature d'une défaillance signalée. Par exemple, l'absence d'une réponse de la part de l'unique système défini peut indiquer une défaillance dans le système cible ou dans l'un des interfaces du groupe IPMP. En revanche, si un seul système entre plusieurs systèmes cible ne répond pas à une sonde, alors la défaillance l'échec est probablement dans le système cible plutôt que dans le groupe IPMP lui-même.

Le démon `in.mpathd` détermine les systèmes cible à analyser de façon dynamique. Tout d'abord, le démon recherche dans la table de routage les systèmes cible sur le même sous-réseau que les adresses de test associées aux interfaces du groupe IPMP. Si de telles cibles sont trouvées, le démon les utilise comme cibles pour les tests. Si aucune cible n'est trouvée sur le

même sous-réseau, le démon envoie des paquets de multidiffusion pour tester les hôtes voisins sur la liaison. Le paquet de multidiffusion est envoyé à l'adresse de multidiffusion de tous les hôtes (`224.0.0.1` dans IPv4 et `ff02::1` dans IPv6) pour déterminer les hôtes à définir en tant que systèmes cible. Les cinq premiers hôtes qui répondent aux paquets d'écho sont sélectionnés en tant que cibles pour les sondes. Si le démon ne parvient pas à trouver des routeurs ou des hôtes ayant répondu aux tests de multidiffusion, le démon ne peut pas assurer la détection de défaillance basée sur sonde. Dans ce cas, la commande `ipmpstat -i` indique que la sonde présente l'état `unknown`.

Vous pouvez établir de façon explicite à l'aide de routes d'hôte la liste des systèmes cible que le démon `in.mpathd` doit utiliser. Pour obtenir des instructions, reportez-vous à “Configuration de la détection de défaillance basée sur sonde” à la page 86

Détection de défaillance basée sur sonde sans utiliser d'adresses de test

Sans adresse de test, cette méthode est implémentée à l'aide de deux types de sondes :

- **Sondes ICMP**

  Les sondes ICMP sont envoyées par les interfaces actives du groupe IPMP pour tester les cibles définies dans la table de routage. Une interface *active* désigne une interface sous-jacente qui peut recevoir des paquets IP entrants envoyés à l'adresse de couche de liaison (L2) de l'interface. La sonde ICMP utilise l'adresse de données comme adresse source de la sonde. Si la sonde ICMP atteint sa cible et obtient une réponse à partir de la cible, alors l'interface active est opérationnelle.
- **Tests transitifs**

  Les sondes transitives sont envoyées par d'autres interfaces du groupe IPMP pour tester l'interface active. Une interface alternative est une interface sous-jacente qui n'est ne reçoit pas activement des paquets IP entrants.

  Par exemple, prenons le cas d'un groupe IPMP composé de quatre interfaces sous-jacentes. Dans cette configuration, les paquets sortants peuvent utiliser toutes les interfaces sous-jacentes. Cependant, les paquets entrants peuvent uniquement être reçus par l'interface à laquelle l'adresse de données est liée. Les trois autres interfaces sous-jacentes qui ne peuvent pas recevoir les paquets entrants sont les interfaces *alternatives*.

  Si une interface alternative réussit à envoyer une sonde à une interface active et à recevoir une réponse, alors l'interface active est fonctionnelle, et par déduction, l'interface alternative qui a envoyé la sonde également.

Remarque - Dans Oracle Solaris, la détection de défaillance basée sur sonde fonctionne avec des adresses de test. Pour sélectionner la détection de défaillance basée sur sonde sans adresse de test, il faut activer manuellement le test transitif. Pour obtenir des instructions, reportez-vous à “La sélection d'un Failure Detection Method” à la page 87.

Défaillance de groupe

Une *défaillance de groupe* survient lorsque la totalité des interfaces d'un groupe IPMP sont défaillantes au même moment. Dans ce cas, aucune interface sous-jacente n'est utilisable. De même, lorsque l'ensemble des systèmes cibles tombe en panne au même moment et que la détection de défaillance basée sur sonde est activée, le démon `in.mpathd` vide tous ses systèmes cibles et test de nouveaux systèmes cibles.

Dans un groupe IPMP qui n'a aucune adresse de test, une interface unique qui peut tester l'interface active est désignée comme *prober*. Cette interface désignée porte les indicateurs `FAILED` et `PROBER`. L'adresse de données est liée à cette interface, ce qui lui permet de poursuivre le test de la cible pour détecter une récupération.

Détection de défaillance basée sur les liaisons

La détection de défaillance basée sur les liaisons est toujours activée, à condition que l'interface prenne ce type de détection en charge.

Pour déterminer si une interface tierce prend en charge la détection de défaillance basée sur les liaisons, utilisez la commande `ipmpstat -i`. Si la sortie d'une interface donnée indique l'état `unknown` dans la colonne `LINK`, l'interface ne prend pas en charge la détection de défaillance basée sur les liaisons. Reportez-vous à la documentation du fabricant pour obtenir des informations plus spécifiques sur le périphérique.

Les pilotes réseau qui prennent en charge la détection de défaillance basée sur les liaisons contrôlent l'état de la liaison de l'interface et notifient le sous-système de mise en réseau en cas de changement d'état. En fonction de la situation, le sous-système de mise en réseau définit ou efface alors l'indicateur `RUNNING` de cette interface. Si le démon `in.mpathd` détecte que l'indicateur `RUNNING` de l'interface a été effacé, il déclenche immédiatement une défaillance de l'interface.

Détection de défaillance et fonction de groupe anonyme

IPMP prend en charge la détection de défaillance dans un groupe anonyme. Par défaut, IPMP surveille l'état uniquement d'interfaces qui appartiennent à des groupes IPMP. Cependant, le démon IPMP peut être configuré de façon à également assurer le suivi du statut des interfaces qui n'appartiennent à aucun groupe IPMP. Par conséquent, ces interfaces sont considérées comme faisant partie d'un *anonymous group*. Lorsque vous exécutez la commande`ipmpstat`, `-le` groupe anonyme g est affiché sous forme de double-tirets (`--`). Dans les groupes anonymes, les adresses de données des interfaces fonctionnent également comme adresse de test. Comme

ces interfaces n'appartiennent pas à un groupe IPMP nommé, ces adresses sont visibles pour les applications. Pour activer le suivi des interfaces qui ne font pas partie d'un groupe IPMP, reportez-vous à la section “Configuration du comportement du démon IPMP ” à la page 89.

Détection de réparation d'interface physique

Le *temps de détection de réparation* est le double du temps de détection de défaillance. Le temps de détection de défaillance par défaut est de 10 secondes. Par conséquent, le temps détection de réparation par défaut est de 20 secondes. Dès qu'une interface défaillante porte de nouveau l'indicateur `RUNNING` et que la méthode de détection de défaillance la considère comme réparée, le démon `in.mpathd` efface l'indicateur `FAILED` de l'interface. L'interface réparée est redéployée en fonction du nombre d'interfaces actives que l'administrateur a définies au départ.

Lorsqu'une interface sous-jacente tombe en panne et que la détection de défaillance basée sur sonde est utilisée, le démon `in.mpathd` poursuit le test, soit par le biais du "prober" désigné si aucune adresse de test n'est configurée, soit à l'aide de l'adresse de test de l'interface.

Au cours d'une réparation d'interface, la façon dont se déroule le processus de récupération dépend de la façon dont l'interface défaillante était à l'origine configuré comme suit :

- Si l'interface défaillante était active à l'origine, l'interface réparée reprend son état actif initial. L'interface de secours qui a pris le relais lors de la défaillance repasse à l'état standby si suffisamment d'interfaces sont actives dans le groupe, tel que défini par l'administrateur système.

Remarque - Une exception à cette règle se présente si l'interface active réparée est également configurée en mode `FAILBACK=no`. Pour plus d'informations, reportez-vous à la section “Mode `FAILBACK=no`” à la page 67.

- Si l'interface défaillante était à l'origine une interface de secours, l'interface réparée revient à son état de secours d'origine, à condition que le groupe IPMP reflète le nombre initial d'interfaces actives. Autrement, l'interface de secours devient une interface active.

Pour une présentation graphique du comportement de la fonctionnalité de chemins d'accès multiples sur réseau IP pendant la panne et la réparation d'une interface, reportez-vous à “Fonctionnement de la fonctionnalité de chemins d'accès multiples sur réseau IP (IPMP)” à la page 56.

Mode `FAILBACK=no`

Par défaut, les interfaces actives qui ont subi une défaillance redeviennent automatiquement des interfaces actives dans le groupe IPMP après leur réparation. Ce comportement est

contrôlé par la valeur du paramètre `FAILBACK` dans le fichier de configuration du démon `in.mpathd`. Cependant, même les perturbations insignifiantes, qui se produisent lorsque les adresses de données sont remappées pour réparer des interfaces, peuvent ne pas être acceptable pour certains administrateurs. Les administrateurs peuvent préférer autoriser une interface de secours activée à continuer à jouer le rôle d'interface active. IPMP permet aux administrateurs de contourner le comportement par défaut pour empêcher une interface de devenir automatiquement active après sa réparation. Ces interfaces doivent être configurées dans le mode `FAILBACK=no`. Pour connaître les procédures connexes, reportez-vous à la section “Configuration du comportement du démon IPMP ” à la page 89.

Lorsqu'une interface active en mode `FAILBACK=no` subit une panne et est ensuite réparée, le démon `in.mpathd` restaure la configuration IPMP comme suit :

- Le démon conserve l'état `INACTIVE` de l'interface, à condition que le groupe IPMP reflète la configuration d'origine des interfaces actives.
- Si, au moment de la réparation, la configuration IPMP ne reflète pas la configuration d'origine du groupe des interfaces actives, l'interface réparée est redéployée comme interface active, malgré l'état `FAILBACK=no`.

Remarque - Le mode `FAILBACK=NO` est défini pour l'ensemble du groupe IPMP, plutôt que sous forme de paramètre réglable par interface.

IPMP et reconfiguration dynamique

La fonctionnalité DR (Dynamic Reconfiguration, reconfiguration dynamique) permet de reconfigurer le matériel système, notamment les interfaces, lorsque le système est en cours d'exécution. La reconfiguration dynamique peut être utilisée uniquement sur les systèmes qui prennent en charge cette fonctionnalité. Sur les systèmes qui prennent en charge la reconfiguration dynamique, IPMP est intégré à la structure Reconfiguration Coordination Manager (RCM, gestionnaire de coordination de reconfiguration). Par conséquent, vous pouvez en toute sécurité connecter, déconnecter ou reconnecter les cartes réseau et le RCM assure la gestion de la reconfiguration dynamique des composants système. Vous pouvez par exemple joindre, raccorder puis ajouter de nouvelles interfaces aux groupes IPMP existants. Une fois ces interfaces configurées, elles sont immédiatement disponibles pour l'utilisation par IPMP.

Toutes les demandes de déconnexion des cartes d'interface réseau sont d'abord vérifiées afin de garantir des connexions ininterrompues. Ainsi, par défaut, il est impossible de déconnecter une carte réseau qui ne fait pas partie d'un groupe IPMP. La déconnexion est également impossible dans le cas d'une carte d'interface réseau qui contient les seules interfaces en état de fonctionnement d'un groupe IPMP. Cependant, si vous devez impérativement retirer le composant système, l'option `-f` de la commande `cfgadm` vous permet de contourner ce comportement, comme indiqué dans la page de manuel `cfgadm`(1M).

Si les vérifications sont satisfaisantes, le démon `in.mpathd` définit l'indicateur `OFFLINE` sur l'interface. Toutes les adresses de test sur les interfaces ne sont pas configurées. Ensuite, la carte d'interface réseau est démontée du système.

En cas d'échec de l'une de ces étapes, ou de défaillance de la reconfiguration dynamique ou autre matériel sur le même composant système, l'état d'origine de la configuration précédente est restauré. Dans ce cas, le message d'erreur suivant est consignée :

```
"IP: persistent configuration is restored for <ifname>"
```

Dans le cas contraire, la demande de déconnexion est traitée. Vous pouvez retirer le composant du système et aucune connexion existante n'est interrompue.

Remarque - Lors du remplacement de cartes réseau, vérifiez qu'elles sont du même type (Ethernet, par exemple). Dès que la carte réseau est remplacée, les configurations d'interface IP persistantes s'appliquent à cette carte.

♦ ♦ ♦ CHAPITRE 3

Administration d'IPMP

Ce chapitre décrit comment gérer des groupes d'interfaces avec IPMP dans la version Oracle Solaris. Les tâches dans ce chapitre expliquent comment configurer IPMP à l'aide de la commande `ipadm` qui remplace la commande `ifconfig` utilisée pour configurer IPMP (multipathing sur réseau IP) dans Oracle Solaris 10. Pour en savoir plus sur la façon dont ces deux commandes sont mappées les unes vers les autres, reportez-vous à " Comparaison de la commande ifconfig et de la commande ipadm " du manuel " Transition d'Oracle Solaris 10 vers Oracle Solaris 11.2 ".

Pour obtenir une explication détaillée des modifications contenues dans le modèle, reportez-vous à "Nouveautés dans IPMP" à la page 49.

Ce chapitre aborde les sujets suivants :

Configuration de groupes IPMP

Les informations suivantes décrivent le fonctionnement des groupes de planification et de configuration IPMP. La présentation du Chapitre 2, IPMP d'administration sur décrit l'implémentation d'un groupe IPMP en tant qu'interface. Par conséquent, les termes *groupe IPMP* et *interface IPMP* sont utilisés de façon interchangeable.

Cette section comprend les tâches suivantes :

▼ Planification pour un groupe IPMP

La procédure suivante inclut les tâches de planification et les informations requise à collecter préalablement à la configuration du groupe IPMP. Vous n'avez pas besoin d'effectuer les tâches suivantes dans l'ordre séquentiel.

La configuration IPMP dépend des exigences de votre réseau d'entreprise pour gérer le type de trafic hébergé sur le système. La fonctionnalité de chemins d'accès multiples sur réseau IP répartit les paquets sortants sur l'ensemble des interfaces du groupe IPMP et améliore ainsi le débit du réseau. Toutefois, pour une connexion TCP donnée, le trafic entrant suit généralement un seul chemin d'accès physique pour minimiser le risque de traiter des paquets hors-service.

Par conséquent, si le réseau traite un volume élevé de trafic sortant, la configuration d'un grand nombre d'interfaces dans un groupe IPMP peut améliorer les performances du réseau. En revanche, si le système gère un fort trafic entrant, le nombre d'interfaces dans le groupe n'améliore pas nécessairement les performances en répartissant la charge de trafic. Toutefois, le fait de disposer de plus d'interfaces sous-jacentes permet de garantir la disponibilité du réseau lors d'une défaillance d'interface.

Remarque - Vous ne devez configurer qu'un seul groupe IPMP pour chaque sous-réseau ou domaine de diffusion L2. Pour plus d'informations, reportez-vous à la section “Règles d'utilisation de la fonctionnalité de chemins d'accès multiples sur réseau IP” à la page 52.

---

1. **Déterminez la configuration IPMP générale qui correspond à vos besoins.**

   Reportez - vous aux informations contenues dans la synthèse des tâches de la procédure ci-dessous , IPMP, qui les aideront à déterminer quelle configuration à utiliser.

2. **(SPARC uniquement) Vérifiez que chaque interface du groupe possède une adresse MAC unique.**

   Pour configurer une adresse MAC unique pour chaque interface du système, reportez-vous à “ Garantie de l’unicité de l’adresse MAC de chaque interface ” du manuel “ Configuration et administration des composants réseau dans Oracle Solaris 11.2 ”

3. **Vérifiez que le même ensemble de modules STREAMS est configuré et empilé sur toutes les interfaces du groupe IPMP.**

   Afin d'obtenir des instructions et la syntaxe de commande pour utiliser IPMP, reportez-vous à “Règles d'utilisation de la fonctionnalité de chemins d'accès multiples sur réseau IP” à la page 52.

4. **Respectez le même format d'adressage IP sur toutes les interfaces du groupe IPMP.**

   Si une interface est configurée pour IPv4, vous devez alors configurer toutes les interfaces du groupe IPMP pour IPv4. De même, si vous ajoutez l'adressage IPv6 sur une interface, vous devez alors configurer toutes les interfaces du groupe IPMP pour prendre en charge IPv6.

5. **Déterminez le type de détection de défaillance que vous souhaitez implémenter.**

   Par exemple, si vous voulez implémenter la détection de défaillance basée sur sonde, vous devez configurer les adresses de test sur les interfaces sous-jacentes. Reportez-vous à “Détection de défaillance dans IPMP” à la page 63.

6. **Assurez-vous que toutes les interfaces du groupe IPMP sont connectées au même réseau local.**

   Par exemple, vous pouvez configurer les commutateurs Ethernet sur le même sous-réseau IP dans un groupe IPMP. Vous pouvez configurer le nombre d'interfaces de votre choix dans un groupe IPMP.

   Remarque - Vous pouvez également configurer un groupe IPMP à interface unique, par exemple, si le système ne dispose que d'une interface physique. “Types de configurations d'interface IPMP” à la page 55

7. **Assurez-vous que le groupe IPMP ne contient pas d'interfaces avec différents types de média réseau.**

   Les interfaces regroupées doivent être du même type. Par exemple, vous ne pouvez pas combiner des interfaces Ethernet et Token Ring dans un groupe IPMP. En outre, vous ne pouvez pas combiner un bus d'interfaces Token avec des interfaces ATM (Asynchronous Transfer Mode, mode de transfert asynchrone) dans le même groupe IPMP.

8. **Pour la fonctionnalité de chemins d'accès multiples sur réseau IP avec interfaces ATM, configurez les interfaces ATM en mode d'émulation LAN.**

   IPMP n'est pas pris en charge sur les interfaces utilisant la technologie Classical IP over ATM, comme indiqué dans les documents IETF RFC 1577 (http://www.rfc-editor.org/rfc/rfc1577.txt) et IETF RFC 2225 (http://www.rfc-editor.org/rfc/rfc2225.txt).

▼ Configuration d'un groupe IPMP utilisant le protocole DHCP

Vous pouvez configurer un groupe IPMP à interface avec plusieurs interfaces actives-actives ou Interfaces actives-de secours. “Types de configurations d'interface IPMP” à la page 55 La procédure ci-dessous indique comment configurer un groupe IPMP actif-de secoursà l'aide du protocole DHCP.

Avant de commencer

Avant d'effectuer les procédures ci-dessous , procédez comme suit :

- Assurez-vous que les interfaces IP qui seront dans le groupe IPMP potentiel ont été correctement configurées sur les liaisons de données du système. Pour plus d'informations sur les procédures, reportez-vous à “ Configuration et administration des composants réseau

dans Oracle Solaris 11.2 ” Vous pouvez créer une interface IPMP même si vous n'avez pas créé les interfaces IP sous-jacente . En revanche, les configurations ultérieures de l'interface IPMP échouent si des interfaces IP sous-jacentes n'existent pas.

- En outre, si vous disposez d'un système SPARC, associez une adresse MAC unique à chaque interface. Reportez-vous à “ Garantie de l’unicité de l’adresse MAC de chaque interface ” du manuel “ Configuration et administration des composants réseau dans Oracle Solaris 11.2 ”.
- Enfin, si vous utilisez DHCP, assurez-vous que les interfaces sous-jacentes ont des *infinite leases* (baux infinis). Autrement, en cas de défaillance d'un groupe IPMP, les adresses de test expirent et le démon `in.mpathd` désactive la détection de défaillance basée sur sonde. Après quoi, la détection de défaillance basée sur les liaisons est mise en oeuvre. Si la détection de défaillance basée sur les liaisons découvre que l'interface fonctionne, le démon peut incorrectement signaler que l'interface a été réparée. Pour plus d'informations sur la configuration DHCP, reportez-vous à “ Utilisation de DHCP dans Oracle Solaris 11.2 ”.

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Créez une interface IPMP.**

   `#` `ipadm create-ipmp` *ipmp-interface*

   Remplacez *ipmp-interface* par le nom de l'interface IPMP. Vous pouvez attribuer n'importe quel nom significatif à l'interface IPMP. Comme n'importe quelle interface IP, le nom est constitué d'une chaîne de caractères et d'un chiffre (`ipmp0`, par exemple).

3. **Au besoin, créez les interfaces IP sous-jacentes.**

   `#` `ipadm create-ip` *under-interface*

   où *under-interface* fait référence à l'interface IP que vous allez ajouter au groupe IPMP.

4. **Ajoutez des interfaces IP sous-jacentes qui contiendront des adresses de test au groupe IPMP.**

   `#` `ipadm add-ipmp -i` *under-interface1* **`[-i`** *under-interface2* **`...]`** *ipmp-interface*

   Vous pouvez ajouter autant d'interfaces IP au groupe IPMP qu'en compte le système.

5. **Faites configurer et gérer les adresses de données par DHCP sur l'interface IPMP.**

   `#` `ipadm create-addr -T dhcp` *ipmp-interface*

   L'étape précédente permet d'associer l'adresse fournie par le serveur DHCP à un objet d'adresse. L'objet d'adresse identifie de manière unique l'adresse IP en respectant le format *interface*/*address-type* (`ipmp0/v4`, par exemple). Pour plus d'informations, reportez-vous à “ Configuration d’une interface IPv4 ” du manuel “ Configuration et administration des composants réseau dans Oracle Solaris 11.2 ”.

6. **Si la détection de défaillance basée sur sonde est activée avec des adresses de test, configurez DHCP pour gérer ces adresses de test sur les interfaces sous-jacentes.**

   ```
   # ipadm create-addr -T dhcp under-interface
   ```

   L'objet d'adresse créé automatiquement à l'étape 6 respecte le format *under-interface*/*address-type*, `net0/v4`, par exemple.

7. **(Facultatif) Répétez la procédure de l 'Step 6 pour chaque interface sous-jacente du groupe IPMP.**

Exemple 3-1 Configuration d'un groupe IPMP avec DHCP

Cet exemple montre comment configurer un groupe IPMP active-de secours sur le serveur DHCP et est basé sur le scénario suivant :

- Trois interfaces sous-jacentes `net0`, `net1` et `net2` sont configurées dans un groupe IPMP.
- L'interface IPMP `ipmp0` porte le même nom que le groupe IPMP.
- `net2` est désignée comme l'interface de secours.
- Toutes les interfaces sous-jacentes se voient attribuer des adresses de test.

La création initiale de l'interface IPMP.

```
# ipadm create-ipmp ipmp0
```

Sont créées des interfaces IP sous-jacente et à ajouter à l'interface IPMP.

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ip net2

# ipadm add-ipmp -i net0 -i net1 -i net2 ipmp0
```

Les adresses IP DHCP gérée par la sont affectées à l'interface IPMP. Les adresses IP affectées à l'interface IPMP sont des adresses de données. Dans cet exemple, l'interface IPMP possède deux adresses de données.

```
# ipadm create-addr -T dhcp ipmp0
ipadm: ipmp0/v4
# ipadm create-addr -T dhcp ipmp0
ipadm: ipmp0/v4a
```

En mode de gestion de DHCP ensuite IP, les adresses sont affectées aux interfaces IP sous-jacente IPMP le groupe IPMP. Les adresses IP affectées aux interfaces sous-jacentes sont des adresses de test à utiliser dans le cadre de la détection de défaillance basée sur sonde.

```
# ipadm create-addr -T dhcp net0
ipadm: net0/v4
# ipadm create-addr -T dhcp net1
ipadm: net1/v4
```

```
# ipadm create-addr -T dhcp net2
ipadm net2/v4
```

Enfin, l'interface `net2` est configurée en tant qu'interface de réserve.

```
# ipadm set-ifprop -p standby=on net2
```

▼ Configuration manuelle d'un groupe IPMP actif-actif

La procédure ci-dessous indique comment configurer manuellement un groupe IPMP actif-actif. Dans cette procédure, les étapes 1 à 4 décrivent la configuration d'un groupe IPMP actif-actif basé sur les liaisons. L'étape 5 indique comment passer d'une configuration basée sur les liaisons à une configuration basée sur les sondes.

Avant de commencer

Assurez-vous que les interfaces IP qui seront dans le groupe IPMP potentiel ont été correctement configurées sur les liaisons de données du système. Pour plus d'informations, reportez-vous à " Configuration d'une interface IPv4 " du manuel " Configuration et administration des composants réseau dans Oracle Solaris 11.2 ". Vous pouvez créer une interface IPMP même si des interfaces IP sous-jacentes n'existent pas encore. Cependant, les configurations suivantes sur cette interface IPMP échoueront.

En outre, si vous disposez d'un système SPARC, associez une adresse MAC unique à chaque interface. Reportez-vous à " Garantie de l'unicité de l'adresse MAC de chaque interface " du manuel " Configuration et administration des composants réseau dans Oracle Solaris 11.2 ".

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Créez une interface IPMP.**

   ```
   # ipadm create-ipmp ipmp-interface
   ```

   Remplacez *ipmp-interface* par le nom de l'interface IPMP. Vous pouvez attribuer n'importe quel nom significatif à l'interface IPMP. Comme n'importe quelle interface IP, le nom est constitué d'une chaîne de caractères et d'un chiffre (`ipmp0`, par exemple).

3. **Ajoutez des interfaces IP sous-jacentes au groupe.**

   ```
   # ipadm add-ipmp -i under-interface1 [-i underinterface2 ...] ipmp-interface
   ```

   où *under-interface* fait référence à l'interface sous-jacente du groupe IPMP. Vous pouvez ajouter autant d'interfaces IP qu'en compte le système.

   Remarque - Dans un environnement à double pile, si vous placez l'instance IPv4 d'une interface dans un groupe particulier, l'instance IPv6 est automatiquement placée dans ce groupe.

4. **Ajoutez des adresses de données à l'interface IPMP.**

   ```
   # ipadm create-addr -a address ipmp-interface
   ```

   Remplacez *address* par une adresse en notation CIDR, le cas échéant.

   Remarque - Vous n'avez besoin que de l'adresse DNS du nom ou de l'adresse IP du groupe IPMP.

5. **Si la détection de défaillance basée sur sonde est activée avec des adresses de test, ajoutez ces adresses sur les interfaces sous-jacentes.**

   ```
   # ipadm create-addr -a address under-interface
   ```

   Remplacez *address* par une adresse en notation CIDR, le cas échéant. Toutes les adresses IP de test d'un groupe IPMP doivent appartenir au même sous-réseau IP et par conséquent, comprendre le même préfixe réseau.

▼ Configuration d'un groupe IPMP active-de secours.

La procédure suivante explique comment configurer un groupe dans lequel une interface IPMP est conservée en tant qu'interface de réserve. Cette interface est déployée uniquement lorsqu'une interface active dans le groupe échoue.

Pour plus d'informations sur les interfaces de secours, reportez-vous à la section “Types de configurations d'interface IPMP” à la page 55.

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Créez une interface IPMP.**

   ```
   # ipadm create-ipmp ipmp-interface
   ```

   Remplacez *ipmp-interface* par le nom de l'interface IPMP.

3. **Ajoutez des interfaces IP sous-jacentes au groupe.**

   ```
   # ipadm add-ipmp -i under-interface1 [-i underinterface2 ...] ipmp-interface
   ```

   où *under-interface* fait référence à l'interface sous-jacente du groupe IPMP. Vous pouvez ajouter autant d'interfaces IP qu'en compte le système.

   Remarque - Dans un environnement à double pile, si vous placez l'instance IPv4 d'une interface dans un groupe IPMP particulier, l'instance IPv6 est automatiquement placée dans ce groupe.

4. **Ajoutez des adresses de données à l'interface IPMP.**

   ```
   # ipadm create-addr -a address ipmp-interface
   ```

   Remplacez *address* par une adresse en notation CIDR, le cas échéant.

5. **Si la détection de défaillance basée sur sonde est activée avec des adresses de test, ajoutez ces adresses sur les interfaces sous-jacentes.**

   ```
   # ipadm create-addr -a address under-interface
   ```

   Remplacez *address* par une adresse en notation CIDR, le cas échéant. Toutes les adresses IP de test d'un groupe IPMP doivent appartenir au même sous-réseau IP et par conséquent, comprendre le même préfixe réseau.

6. **Configurez l'une des interfaces sous-jacentes en tant qu'interface de secours.**

   ```
   # ipadm set-ifprop -p standby=on -m ip under-interface
   ```

Exemple 3-2 Configuration d'un groupe IPMP active-de secours

L'exemple suivant montre comment créer une configuration IPMP active-de réserve.

Tout d'abord, l'interface IPMP est créé.

```
# ipadm create-ipmp ipmp0
```

Les interfaces IP sous-jacentes sont ensuite créer et à ajouter à l'interface IPMP.

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ip net2

# ipadm add-ipmp -i net0 -i net1 -i net2 ipmp0
```

Les adresses IP sont affectées à l'interface IPMP. Les adresses IP affectées à l'interface IPMP sont des adresses de données. Dans cet exemple, l'interface IPMP possède deux adresses de données.

```
# ipadm create-addr -a 192.168.10.10/24 ipmp0
ipadm: ipmp0/v4
# ipadm create-addr -a 192.168.10.15/24 ipmp0
ipadm: ipmp0/v4a
```

L'adresse IP de cet exemple inclut une propriété `prefixlen` exprimée sous forme de nombre décimal. La partie `prefixlen` de l'adresse IP indique le nombre de bits contigus complètement à gauche de l'adresse, qui correspond au masque de réseau IPv4 ou au préfixe IPv6 de l'adresse. Les bits de poids faible restants définissent la partie hôte de l'adresse. Lorsque la propriété `prefixlen` est convertie en une représentation textuelle de l'adresse, les positions de bits utilisées par la partie réseau sont représentées par une série de 1 et celles utilisées par la partie

hôte par une série de 0. Cette propriété n'est pas prise en charge sur le type d'objet d'adresse `dhcp`. Pour plus d'informations, reportez-vous à la page de manuel `ipadm`(1M).

Les adresses IP sous-jacentes sont ensuite affectés aux interfaces d'IP IPMP le groupe IPMP. Les adresses IP affectées aux interfaces sous-jacentes sont des adresses de test à utiliser dans le cadre de la détection de défaillance basée sur sonde.

```
# ipadm create-addr -a 192.168.10.30/24 net0
ipadm: net0/v4
# ipadm create-addr -a 192.168.10.32/24 net1
ipadm: net1/v4
# ipadm create-addr -a 192.168.10.34/24 net2
ipadm: net2/v4
```

Enfin, l'interface `net2` est configurée en tant qu'interface de réserve.

```
# ipadm set-ifprop -p standby=on net2
```

L'administrateur peut consulter la configuration IPMP en exécutant la commande `ipmpstat`.

```
# ipmpstat -g
GROUP     GROUPNAME   STATE      FDT        INTERFACES
ipmp0     ipmp0       ok         10.00s     net0 net1 (net2)

# ipmpstat -t
INTERFACE  MODE     TESTADDR         TARGETS
net0       routes   192.168.10.30    192.168.10.1
net1       routes   192.168.10.32    192.168.10.1
net2       routes   192.168.10.34    192.168.10.5
```

Maintien du routage lors du déploiement d'IPMP

Lorsque vous configurez un groupe IPMP, l'interface IPMP hérite des adresses IP de ses interfaces sous-jacentes et les utilise en tant qu'adresses de données. Les interfaces sous-jacentes se voient alors attribuer l'adresse IP `0.0.0.0`. Par conséquent, les routes définies sur la base d'interfaces IP données sont perdues si ces interfaces sont ensuite ajoutées à un groupe IPMP.

Une perte de routage lors de la configuration IPMP a fréquemment un rapport avec la route par défaut et se produit lors d'une installation d'Oracle Solaris. Lors de l'installation, vous devez définir une route par défaut pour laquelle vous utilisez une interface au sein du système, comme par exemple l'interface principale. Par la suite, vous configurez un groupe IPMP en utilisant l'interface sur laquelle vous avez défini la route par défaut. Après la configuration IPMP, le système ne peut plus acheminer de paquets réseau car l'adresse de l'interface a été transférée à l'interface IPMP.

Pour que la route par défaut soit préservée pendant l'utilisation d'IPMP, elle doit être définie sans spécifier d'interface. De cette manière, n'importe quelle interface, y compris l'interface

IPMP, peut être utilisée pour le routage. Le système peut alors continuer à acheminer le trafic réseau.

Remarque - La tâche suivante utilise l'interface primaire en tant qu'exemple selon lequel la route par défaut est définie. Toutefois, le problème d'interruption du routage concerne n'importe quelle interface utilisée pour le routage et qui est, par la suite, intégrée dans un groupe IPMP.

▼ Préservation d'IPMP la route par défaut lors de l'utilisation de

La procédure suivante explique comment préserver la route par défaut lors de la configuration IPMP.

1. **Connectez-vous au système par le biais d'une console.**

   Vous devez effectuer cette procédure via la console. Si vous vous connectez à l'aide des commandes `ssh` ou `telnet`, la connexion est interrompue lors des étapes suivantes.

2. **(Facultatif) Affichez les routes définies dans la table de routage.**

   ```
   # netstat -nr
   ```

3. **Supprimez la route liée à l'interface concernée.**

   ```
   # route -p delete default gateway-address -ifp interface
   ```

4. **Ajoutez la route sans indiquer d'interface.**

   ```
   # route -p add default gateway-address
   ```

5. **(Facultatif) Afficher la redéfini les routes dans la table de routage.**

   ```
   # netstat -nr
   ```

6. **(Facultatif) Si les informations de la table de routage n'ont pas changé, redémarrez le service de routage, puis vérifiez à nouveau les informations figurant dans la table de routage pour vous assurer que les routes ont été redéfinies correctement.**

   ```
   # svcadm restart routing-setup
   ```

Exemple 3-3 Définition de routes pour IPMP

Cet exemple suppose que la route par défaut a été définie pour `net0` au cours de l'installation.

```
# netstat -nr
Routing Table: IPv4
Destination     Gateway       Flags    Ref     Use       Interface
-------------  ------------  --------  -----  -----------  --------
default        10.153.125.1     UG      107    176682262   net0
10.153.125.0   10.153.125.222   U        22    137738792   net0

# route -p delete default 10.153.125.1 -ifp net0
# route -p add default 10.153.125.1

# netstat -nr
Routing Table: IPv4
Destination     Gateway       Flags    Ref     Use       Interface
-------------  ------------  --------  -----  -----------  --------
default        10.153.125.1     UG      107    176682262
10.153.125.0   10.153.125.222   U        22    137738792   net0
```

Administration d'IPMP

Cette section présente les procédures à suivre pour gérer le groupe IPMP créé sur le système.

▼ Ajout d'une interface à un groupe IPMP

Avant de commencer

Assurez - vous que l'interface que vous envisagez d'ajouter au groupe répond à toutes les exigences. Pour obtenir la liste des exigences, reportez-vous à la section “Planification pour un groupe IPMP” à la page 72.

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Si l'interface IP sous-jacente n'existe pas encore, créez l'interface.**

   # **ipadm create-ip** *under-interface*

3. **Ajoutez l'interface IP au groupe IPMP.**

   # **ipadm add-ipmp -i** *under-interface ipmp-interface*

Remplacez *ipmp-interface* par le groupe IPMP auquel ajouter l'interface sous-jacente.

Exemple 3-4 Ajout d'une interface à un groupe IPMP

L'exemple ci-dessous illustre comment ajouter une interface `net4` au groupe IPMP `ipmp0`.

```
# ipadm create-ip net4
# ipadm add-ipmp -i net4 ipmp0
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT       INTERFACES
ipmp0   ipmp0      ok         10.00s    net0 net1 net4
```

▼ Retrait d'une interface d'un groupe IPMP

1. **Connectez-vous en tant qu'utilisateur `root`.**
2. **Retirez une ou plusieurs interfaces du groupe IPMP.**

   `# ipadm remove-ipmp -i` *under-interface*`[ -i` *under-interface* `...]` *ipmp-interface*

   Remplacez *under-interface* par l'interface IP à retirer du groupe IPMP et *ipmp-interface* par le groupe IPMP duquel retirer l'interface sous-jacente.

   Vous pouvez retirer autant d'interfaces sous-jacentes que vous le souhaitez dans une seule commande. Le retrait de toutes les interfaces sous-jacentes ne supprime pas l'interface IPMP. Celle-ci reste sous la forme d'un groupe ou d'une interface IPMP vide.

Exemple 3-5 Retrait d'une interface d'un groupe IPMP

L'exemple suivant montre comment supprimer l'interface `net4` du groupe IPMP `ipmp0`.

```
# ipadm remove-ipmp net4 ipmp0
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT       INTERFACES
ipmp0   ipmp0      ok         10.00s    net0 net1
```

▼ Ajout d'adresses IP à un groupe IPMP

Pour ajouter à un groupe IPMP des adresses IP, utilisez la commande. `ipadm create-addr`. Notez qu'une adresse IP peut être soit une adresse de données, soit une adresse de test dans IPMP. Une adresse de données est ajoutée à une interface, tandis que IPMP une adresse de test

est ajoutée à une interface sous-jacente de l'interface IPMP. La procédure ci-dessous indique comment ajouter des adresses IP, qu'il s'agisse d'adresses de test ou de données.

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Ajoutez les adresses IP à un groupe IPMP.**

   - **Ajouter des adresses de données à un groupe IPMP comme suit :**

     ```
     # ipadm create-addr -a address ipmp-interface
     ```

     Un objet d'adresse est automatiquement associé à l'adresse IP que vous venez de créer. Un objet d'adresse correspond à l'identifiant unique d'une adresse IP. L'objet d'adresse respecte la convention de dénomination *interface*/*random-string*. Ainsi, le nom d'un objet d'adresse de données inclut l'interface IPMP.

   - **Pour ajouter des adresses de test à l'interface sous-jacente d'un groupe IPMP, entrez la commande suivante :**

     ```
     # ipadm create-addr -a address under-interface
     ```

     Un objet d'adresse est automatiquement associé à l'adresse IP que vous venez de créer. Un objet d'adresse correspond à l'identifiant unique d'une adresse IP. L'objet d'adresse respecte la convention de dénomination *interface*/*random-string*. Ainsi, le nom d'un objet d'adresse de test inclut l'interface sous-jacente.

▼ Suppression d'adresses IP d'une interface IPMP

Pour supprimer des adresses IP à partir d'un groupe IPMP, utilisez la commande `ipadm delete-addr`. Notez que les adresses de données sont hébergées sur l'interface IPMP et les adresses de test sur les interfaces sous-jacentes dans IPMP. La procédure ci-dessous indique comment retirer des adresses IP, qu'il s'agisse d'adresses de test ou de données.

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Déterminez les adresses à supprimer.**

   - **Afficher la liste des adresses de données comme suit :**

     ```
     # ipadm show-addr ipmp-interface
     ```

   - **Afficher la liste des adresses de test comme suit :**

     ```
     # ipadm show-addr
     ```

Les adresses de test sont identifiées par les objets d'adresse dont le nom inclut les interfaces sous-jacentes sur lesquelles elles sont configurées.

3. **A partir d'un enlever le groupe IPMP adresses IP.**

   - **Supprimez des adresses de données comme suit :**

     ```
     # ipadm delete-addr addrobj
     ```

     Remplacez *addrobj* par l'objet d'adresse de données, qui doit impérativement inclure le nom de l'interface IPMP. Si l'objet d'adresse que vous saisissez ne comporte pas le nom de l'interface IPMP, l'adresse supprimée n'est pas une adresse de données.

   - **Supprimez des adresses de test comme suit :**

     ```
     # ipadm delete-addr addrobj
     ```

     Remplacez *addrobj* par l'objet d'adresse de test, qui doit impérativement inclure le nom de l'interface sous-jacente appropriée pour garantir la suppression de la bonne adresse.

Exemple 3-6 Suppression d'une adresse de test d'une interface

L'exemple ci-dessous reprend la configuration du groupe IPMP actif-de secours `ipmp0` de l'Exemple 3-2, “Configuration d'un groupe IPMP active-de secours ”. Cette commande retire l'adresse de test de l'interface sous-jacente `net1`.

```
# ipadm show-addr net1
ADDROBJ           TYPE     STATE     ADDR
net1/v4           static   ok        192.168.10.30

# ipadm delete-addr net1/v4
```

▼ Déplacement d'une interface d'un groupe IPMP vers un autre

Vous pouvez placer une interface dans un nouveau groupe IPMP lorsque l'interface appartient à un groupe IPMP. Inutile de retirer l'interface du groupe IPMP actuel. Lorsque vous placez l'interface dans un nouveau groupe, l'interface est automatiquement retirée de tout groupe IPMP existant.

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Déplacez l'interface vers un nouveau groupe IPMP.**

   ```
   # ipadm add-ipmp -i under-interface ipmp-interface
   ```

Remplacez *under-interface* par l'interface sous-jacente à déplacer et *ipmp-interface* par l'interface IPMP de destination.

Exemple 3-7 Déplacement d'une interface vers un autre groupe IPMP

Dans l'exemple suivant, les interfaces sous-jacentes du groupe IPMP sont `net0`, `net1` et `net2`. Cet exemple montre comment supprimer l'interface `net0` du groupe IPMP `ipmp0` et placer `net0` dans le groupe IPMP `cs-link1`.

```
# ipadm add-ipmp -i net0 ca-link1
```

▼ Suppression d'un groupe IPMP

Utilisez cette procédure si vous n'avez plus besoin d'un groupe IPMP spécifique.

1. **Connectez-vous en tant qu'utilisateur `root`.**
2. **Identifiez le groupe IPMP et les interfaces IP sous-jacentes à supprimer.**

   ```
   # ipmpstat -g
   ```

3. **Retirez toutes les interfaces IP qui appartiennent actuellement au groupe IPMP.**

   `# ipadm remove-ipmp -i` *under-interface*`[, -i` *under-interface*`, ...]` *ipmp-interface*

   Remplacez *under-interface* par l'interface sous-jacente à retirer et *ipmp-interface* par l'interface IPMP de laquelle retirer l'interface sous-jacente.

   Remarque - Pour supprimer une interface IPMP, aucune interface IP ne doit faire partie du groupe IPMP.

4. **Supprimez l'interface IPMP.**

   `# ipadm delete-ipmp` *ipmp-interface*

   Une fois que vous avez supprimé l'interface IPMP, n'importe quelle adresse IP qui n'est associée à l'interface est supprimée du système.

Exemple 3-8 Suppression d'une interface IPMP

L'exemple de commande ci-dessous supprime l'interface `ipmp0` qui comprend les interfaces IP sous-jacentes `net0` et `net1`.

```
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT        INTERFACES
```

```
ipmp0   ipmp0      ok         10.00s     net0 net1

# ipadm remove-ipmp -i net0 -i net1 ipmp0

# ipadm delete-ipmp ipmp0
```

Configuration de la détection de défaillance basée sur sonde

Cette section s'articule autour des rubriques suivantes :

Détection de défaillance basée sur sonde

La détection de défaillance basée sur sonde nécessite l'utilisation de systèmes cible, comme expliqué dans “Détection de défaillance basée sur sonde” à la page 64. Pendant l'identification des cibles concernées par la détection de défaillance basée sur sonde, le démon `in.mpathd` fonctionne en deux modes routeur: *cible routeur* ou *cible multidiffusion*. En mode cible routeur, le démon teste les cibles définies dans la table de routage. Si aucune cible n'est définie, le démon s'exécute en mode cible multidiffusion, où les paquets de multidiffusion sont envoyés tester les hôtes voisins sur le réseau local.

Mieux vaut définir les systèmes cible que le démon `in.mpathd` doit tester. Pour certains groupes IPMP, le routeur par défaut est suffisant comme cible. Cependant, pour certains groupes IPMP, il est nécessaire de configurer des cibles spécifiques pour la détection de défaillance basée sur sonde. Pour spécifier les cibles, définissez des routes hôte dans la table de routage comme cibles de sonde. Les routes hôte configurées dans la table de routage figurent dans la liste avant le routeur par défaut. IPMP utilise les routes hôte définies explicitement pour la sélection de cibles. Par conséquent, vous devez définir des routes hôte afin de configurer des cibles de sondes plutôt que d'utiliser le routeur par défaut.

Pour définir des routes hôte dans la table de routage, utilisez la commande `route`. Vous pouvez utiliser l'option `-p` avec cette commande pour ajouter des routes permanentes. Par exemple, la commande `route -p add` ajoute une route qui reste dans la table de routage même après la réinitialisation du système. L'option `-p` vous permet d'ajouter des routes persistantes sans rédiger de scripts spéciaux pour les recréer à chaque démarrage du système. Pour mieux utiliser

la détection de défaillance basée sur sonde, assurez-vous que vous avez défini plusieurs cibles pour recevoir les sondes.

La commande `route` fonctionne sur les routes IPv4 (par défaut) et IPv6. Pour réaliser des opérations sur les routes IPv6, tapez l'option `-inet6` immédiatement à la suite de la commande `route` dans la ligne de commande.

La procédure “Spécification manuelle de systèmes cible pour la détection de défaillance basée sur sonde” à la page 88 présente la syntaxe exacte pour ajouter des routes persistantes vers les cibles concernées par la détection de défaillance basée sur sonde. Pour plus d'informations sur les options qui peuvent être utilisées avec la commande `route`, reportez-vous à la page de manuel `route`(1M) et à “Maintien du routage lors du déploiement d'IPMP” à la page 79.

Exigences en matière de sélection des cibles concernées par la détection de défaillance basée sur sonde

Consultez la configuration requise suivante pour déterminer les hôtes du réseau les plus adaptés en tant que cibles.

- Assurez-vous que les cibles potentielles sont disponibles et en cours d'exécution. Etablissez la liste de leurs adresses IP.
- Vérifiez que les interfaces cible se trouvent sur le même réseau que le groupe IPMP configuré.
- Le masque de réseau et l'adresse de diffusion des systèmes cible doivent correspondre à ceux du groupe IPMP.
- L'hôte cible doit pouvoir répondre aux demandes ICMP provenant de l'interface qui utilise la détection de défaillance basée sur sonde.

La sélection d'un Failure Detection Method

détection de défaillance basée sur sonde fonctionne à l'aide d'adresses de test, Si le pilote de la carte réseau la prend en charge, la détection de défaillance basée sur les liaisons est également activée automatiquement.

Vous ne pouvez pas désactiver la détection de défaillance basée sur les liaisons si cette méthode est prise en charge par le pilote de la carte réseau. Cependant, vous pouvez sélectionner le type de la détection de défaillance basée sur sonde à implémenter.

Avant de sélectionner une métode de détection par sonde, assurez-vous que vos cibles de sonde respectent les exigences répertoriées dans “Exigences en matière de sélection des cibles concernées par la détection de défaillance basée sur sonde” à la page 87.

Pour n'utiliser que le test transitif, effectuez les opérations suivantes :

1. Activez la propriété IPMP `transitive-probing` à l'aide des commandes SMF.

   ```
   # svccfg -s svc:/network/ipmp setprop config/transitive-probing=true
   # svcadm refresh  svc:/network/ipmp:default
   ```

   Pour plus d'informations sur la configuration de cette propriété, reportez-vous à la page de manuel `in.mpathd`(1M).
2. Retirez toutes les adresses de test existantes précédemment configurées pour le groupe IPMP.

   ```
   # ipadm delete-addr address addrobj
   ```

   Remplacez *addrobj* par l'objet d'adresse, qui doit impérativement contenir le nom de l'interface sous-jacente qui héberge une adresse de test.

Pour utiliser des adresses de test dans le cadre de la détection de défaillance, effectuez les opérations suivantes :

1. Si nécessaire, désactivez le test transitif à l'aide des commandes SMF.

   ```
   # svccfg -s svc:/network/ipmp setprop config/transitive-probing=false
   # svcadm refresh  svc:/network/ipmp:default
   ```
2. Attribuez des adresses de test pour les interfaces sous-jacentes du groupe IPMP.

   ```
   # ipadm create-addr -a address under-interface
   ```

   Remplacez *address* par l'adresse (notation CIDR, le cas échéant) et *under-interface* par l'interface sous-jacente du groupe IPMP.

▼ Spécification manuelle de systèmes cible pour la détection de défaillance basée sur sonde

Cette procédure explique comment ajouter des cibles spécifiques si vous utilisez des adresses de test pour implémenter la détection de défaillance basée sur sonde.

Avant de commencer Vérifiez que les cibles de la sonde répondent aux critères répertoriés à la section “Exigences en matière de sélection des cibles concernées par la détection de défaillance basée sur sonde” à la page 87.

1. **Connectez-vous à l'aide de votre compte utilisateur au système sur lequel vous configurez la détection de défaillance basée sur sonde.**
2. **Ajoutez une route à un hôte spécifique, à utiliser en tant que cible dans le cadre de la détection de défaillance basée sur sonde.**

```
% route -p add -host destination-IP gateway-IP -static
```

où *destination-IP* et *gateway-IP* sont les adresses IPv4 de l'hôte à utiliser en tant que cible. Par exemple, saisissez ce qui suit afin de spécifier le système cible `192.168.10.137` qui se trouve sur le même sous-réseau que les interfaces du groupe IPMP `ipmp0` :

```
% route -p add -host 192.168.10.137 192.168.10.137 -static
```

Cette nouvelle route est configurée automatiquement chaque fois que le système est redémarré. Si vous souhaitez définir une route temporaire vers un système cible pour la détection de défaillance basée sur sonde, n'ajoutez pas l'option `-p`.

3. **Ajoutez les routes vers les hôtes supplémentaires du réseau à utiliser en tant que systèmes cible.**

▼ Configuration du comportement du démon IPMP

Définissez les paramètres système suivants pour les groupes IPMP dans le fichier de configuration IPMP `/etc/default/mpathd` :

- `FAILURE_DETECTION_TIME`
- `FAILBACK`
- `TRACK_INTERFACES_ONLY_WITH_GROUPS`

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Modifiez le fichier `/etc/default/mpathd`.**

```
# pfedit /etc/default/mpathd
```

Reportez-vous à la page de manuel `pfedit`(1M) pour obtenir des instructions.

Modifiez la valeur par défaut d'au moins l'une des trois paramètres suivants :

- Saisissez la nouvelle valeur du paramètre `FAILURE_DETECTION_TIME` comme suit :

  ```
  FAILURE_DETECTION_TIME=n
  ```

  où *n* correspond à la durée en secondes nécessaire aux sondes ICMP pour la détection d'une éventuelle défaillance d'interface. La valeur par défaut est de 10 secondes.

- Saisissez la nouvelle valeur du paramètre `FAILBACK` comme suit :

  ```
  FAILBACK=[yes | no]
  ```

  `Yes` — Est la valeur par défaut pour le comportement de basculement IPMP. En cas de détection de réparation d'une interface défaillante, l'accès

réseau est rétabli à l'aide de l'interface réparée, tel que décrit dans la section “Détection de réparation d'interface physique” à la page 67.

`No`
: Indique que le trafic de données n'est pas renvoyé sur une interface réparée. En cas de détection d'une interface réparée, l'indicateur `INACTIVE` est défini pour celle-ci. L'indicateur indique qu'il est actuellement impossible d'utiliser l'interface pour le trafic de données. Il est cependant possible de l'utiliser pour le trafic de sondes.

  Imaginons par exemple que le groupe IPMP `ipmp0` se compose de deux interfaces (`net0` et `net1`). Dans le fichier `/etc/default/mpathd`, le paramètre `FAILBACK=no` est défini. Si `net0` échoue, elle est marquée comme étant `FAILED` et devient inutilisable. Après réparation, l'interface est marquée comme étant `INACTIVE` et reste inutilisables en raison de la valeur `FAILBACK=no`.

  En cas de défaillance de `net1`, si seule l'interface `net0` présente l'état `INACTIVE`, l'indicateur `INACTIVE` de `net0` est effacé et l'interface redevient opérationnelle. Si le groupe IPMP dispose d'autres interfaces qui sont également dans l'état `INACTIVE`, alors n'importe laquelle de ces interfaces `INACTIVE`, et pas nécessairement `net0`, peut être effacée et devenir utilisable lorsque `net1` échoue.

- Saisissez la nouvelle valeur du paramètre `TRACK_INTERFACES_ONLY_WITH_GROUPS` comme suit :

  ```
  TRACK_INTERFACES_ONLY_WITH_GROUPS=[yes | no]
  ```

  `Yes`
  : Correspond au comportement par défaut d'IPMP. Cette valeur permet à IPMP d'ignorer les interfaces réseau qui ne sont pas configurées dans un groupe IPMP.

  `No`
  : Définit la détection de défaillance et de réparation pour *toutes* les interfaces réseau, qu'elles soient configurées ou non dans un groupe IPMP. Cependant, lorsqu'une défaillance ou une réparation est détectée sur une interface non configurée dans un groupe IPMP, aucune action n'est déclenchée dans IPMP pour maintenir les fonctions réseau de cette interface. Par conséquent, la valeur `no` s'avère utile uniquement pour signaler les défaillances et n'améliore pas directement la disponibilité du réseau.

    Pour plus d'informations sur ce paramètre et la fonction de groupe anonyme, reportez-vous à la section “Détection de défaillance et fonction de groupe anonyme ” à la page 66.

3. **Redémarrez le démon `in.mpathd`.**

   ```
   # pkill -HUP in.mpathd
   ```

Le démon redémarre avec les nouvelles valeurs de paramètre en vigueur.

Surveillance des informations IPMP

Les exemples suivants illustrent l'utilisation de la commande `ipmpstat` afin de surveiller différents aspects des groupes IPMP qui se trouvent sur le système. Vous pouvez observer l'état d'un groupe IPMP dans son ensemble ou de ses interfaces IP sous-jacentes. Vous pouvez également vérifier la configuration des adresses de données et de test d'un groupe IPMP. Vous pouvez également utiliser cette même commande pour obtenir des informations sur la détection de défaillance. Pour plus d'informations, reportez-vous à la page de manuel `ipmpstat`(1M) .

Lorsque vous exécutez la commande `ipmpstat`, les champs les plus significatifs qui tiennent dans 80 colonnes sont affichés par défaut. Dans la sortie, tous les champs qui sont spécifiques à l'option que vous pouvez utiliser avec la commande `ipmpstat` sont affichés, sauf dans le cas où `ipmpstat` est utilisé avec l'option `-p`.

Par défaut, les noms d'hôte sont affichés dans la sortie (et non les adresses IP numériques), à condition que ces noms existent. Pour répertorier les adresses IP numériques dans la sortie, ajoutez l'option `-n` avec d'autres options pour afficher des informations spécifiques sur le groupe IPMP.

Remarque - Dans les exemples suivants, l'exécution de la commande `ipmpstat` ne nécessite pas de privilèges d'administrateur système, sauf indication contraire.

Utilisez la commande `ipmpstat` avec les options suivantes pour afficher les informations souhaitées :

`-g` Affiche des informations sur les groupes IPMP du système. Reportez-vous à la section Exemple 3-9, “Obtention d'informations sur les groupes IPMP”.

`-a` Affiche les adresses de données configurées pour les groupes IPMP. Reportez-vous à la section Exemple 3-10, “Obtention d'informations sur les adresses de données IPMP”.

`-i` Affiche des informations sur les interfaces IP liées à la configuration IPMP. Reportez-vous à la section Exemple 3-11, “Obtention d'informations sur les interfaces IP sous-jacentes d'un groupe IPMP”.

`-t` Affiche des informations sur les systèmes cible par le biais desquels est réalisée la détection de défaillance. Cette option répertorie également

les adresses de test qu'utilise le groupe IPMP. Reportez-vous à la section Exemple 3-12, “Obtention d'informations sur les cibles de sonde IPMP”.

`-p` Affiche des informations sur les sondes en cours d'utilisation dans le cadre de la détection de défaillance. Reportez-vous à la section Exemple 3-13, “Observation des tests IPMP”.

Les exemples décrivent la manière supplémentaires suivants pour afficher les informations relative à la configuration IPMP du système à l'aide de la commande `ipmpstat`.

EXEMPLE 3-9 Obtention d'informations sur les groupes IPMP

L'option `-g` permet d'afficher l'état de plusieurs groupes IPMP au sein du système, avec l'état de leurs interfaces sous-jacentes. Si la détection de défaillance basée sur sonde est activée pour un groupe spécifique, la commande inclut également le temps de détection de défaillance pour ce groupe.

```
% ipmpstat -g
GROUP    GROUPNAME   STATE      FDT        INTERFACES
ipmp0    ipmp0       ok         10.00s     net0 net1
acctg1   acctg1      failed     --         [net3 net4]
field2   field2      degraded   20.00s     net2 net5 (net7) [net6]
```

Les champs figurant dans la sortie indiquent les informations suivantes.

`GROUP` Spécifie le nom de l'interface IPMP. S'il s'agit d'un groupe anonyme, ce champ est vide. Pour plus d'informations sur les groupes anonymes, reportez-vous à la page de manuel `in.mpathd`(1M).

`GROUPNAME` Spécifie le nom du groupe IPMP. Dans le cas d'un groupe anonyme, ce champ est vide.

`STATE` Indique l'état actuel d'un groupe IPMP, à savoir :

- `ok` : indique que toutes les interfaces sous-jacentes du groupe IPMP sont utilisables.
- `degraded` : indique que certaines interfaces sous-jacentes du groupe sont inutilisables.
- `failed` : indique que toutes les interfaces du groupe sont inutilisables.

`FDT` Indique le temps de détection de défaillance, si la détection de défaillance est activée. Si la détection de défaillance est désactivée, ce champ est vide.

`INTERFACES` Spécifie les interfaces sous-jacentes qui appartiennent au groupe IPMP. Dans ce champ, les interfaces actives sont répertoriées en premier, suivies des interfaces inactives et enfin des interfaces inutilisables. L'état d'une interface est précisé selon la manière dont son nom est indiqué :

- *interface* (sans parenthèses ni crochets) : indique une interface active. c'est-à-dire en cours d'utilisation par le système pour envoyer ou recevoir le trafic de données.
- *(interface)* (entre parenthèses) : indique une interface fonctionnelle mais inactive. L'interface n'est pas en cours d'utilisation comme défini par la stratégie d'administration.
- *[interface]* (entre crochets) : indique que l'interface est inutilisable en raison d'une défaillance ou de sa mise hors ligne.

EXEMPLE 3-10 Obtention d'informations sur les adresses de données IPMP

L'option `-a` permet d'afficher les adresses de données et le groupe IPMP auquel appartient chaque adresse. Les informations affichées incluent également les adresses disponibles, selon qu'elles ont été activées ou désactivées par le biais de la commande `ipadm [up-addr/down-addr]`. Vous pouvez également déterminer sur quelle interface entrante ou sortante une adresse peut être utilisée.

```
% ipmpstat -an
ADDRESS         STATE    GROUP      INBOUND     OUTBOUND
192.168.10.10   up       ipmp0         net0    net0 net1
192.168.10.15   up       ipmp0         net1    net0 net1
192.0.0.100     up       acctg1        --          --
192.0.0.101     up       acctg1        --          --
192.168.10.31   up       field2        net2    net2 net7
192.168.10.32   up       field2        net7    net2 net7
192.168.10.33   down     field2        --          --
```

Les champs figurant dans la sortie indiquent les informations suivantes.

`ADDRESS` Spécifie le nom d'hôte ou l'adresse de données si l'option `-n` est utilisée avec l'option `-a`.

`STATE` Indique si l'adresse sur l'interface IPMP est `up`, et donc utilisable, ou `down`, et donc inutilisable.

`GROUP` Spécifie l'interface IPMP qui héberge une adresse de données spécifique. En règle générale, dans Oracle Solaris, le nom de groupe est représenté par l'interface IPMP IPMP.

`INBOUND` Identifie l'interface qui reçoit des paquets pour une adresse donnée. Les informations du champ peuvent changer en fonction des événements externes. Par exemple, si une adresse de données est hors service ou si le groupe IPMP ne contient plus aucune interface IP active, ce champ est vide. Le champ vide indique que le système n'accepte pas les paquets IP qui sont destinés à l'adresse indiquée.

`OUTBOUND` Identifie l'interface qui envoie des paquets qui utilisent une adresse donnée comme adresse source. Comme avec le champ `INBOUND`, le contenu du champ `OUTBOUND` peut varier en fonction d'événements externes. Un champ vide indique que le système n'envoie pas de paquets avec l'adresse source spécifiée. Le champ peut être vide parce que l'adresse est arrêtée ou parce qu'il ne reste pas d'interfaces IP active dans le groupe.

EXEMPLE 3-11 Obtention d'informations sur les interfaces IP sous-jacentes d'un groupe IPMP

L'option `-i` permet d'afficher des informations relatives aux interfaces IP sous-jacentes d'un groupe IPMP.

```
% ipmpstat -i
INTERFACE   ACTIVE   GROUP      FLAGS      LINK       PROBE      STATE
net0        yes      ipmp0      --mb---    up         ok         ok
net1        yes      ipmp0      -------    up         disabled   ok
net3        no       acctg1     -------    unknown    disabled   offline
net4        no       acctg1     is-----    down       unknown    failed
net2        yes      field2     --mb---    unknown    ok         ok
net6        no       field2     -i-----    up         ok         ok
net5        no       filed2     -------    up         failed     failed
net7        yes      field2     --mb---    up         ok         ok
```

Les champs figurant dans la sortie indiquent les informations suivantes.

`INTERFACE` Spécifie chacune des interfaces sous-jacentes des groupes IPMP.

`ACTIVE` Indique si l'interface est opérationnelle et en cours d'utilisation (`yes`) ou pas (`no`).

`GROUP` Spécifie le nom de l'interface IPMP. S'il s'agit d'un groupe anonyme, ce champ est vide. Pour plus d'informations sur les groupes anonymes, reportez-vous à la page de manuel `in.mpathd`(1M).

`FLAGS` Indique l'état de chaque interface sous-jacente, à savoir (combinaison possible) :

- `b` : indique que l'interface est désignée par le système pour recevoir le trafic de diffusion pour le groupe IPMP.
- `d` : indique que l'interface est hors service et par conséquent inutilisable.
- `h` : indique que l'interface physique partage une adresse matérielle physique double avec une autre interface et a été mise hors ligne. L'indicateur `h` signifie que l'interface est inutilisable.

- `i` : précise que l'indicateur `INACTIVE` est défini pour l'interface. Par conséquent, l'interface n'est pas utilisée pour envoyer ou recevoir le trafic de données.
- `m` : indique que l'interface est désignée par le système pour envoyer et recevoir le trafic de multidiffusion IPv4 pour le groupe IPMP.
- `M` : indique que l'interface est désignée par le système pour envoyer et recevoir le trafic de multidiffusion IPv6 pour le groupe IPMP.
- `s` : indique que l'interface est configurée pour être une interface de secours.

`LINK` Indique le statut de la détection de défaillance basée sur les liaisons, à savoir :

- `up` ou `down` indique la disponibilité ou l'indisponibilité d'une liaison.
- `unknown` indique que le pilote ne prend pas en charge la notification indiquant si une liaison est opérationnelle (`up`) ou hors service (`down`) et qu'il ne détecte donc pas les changements d'état de liaison.

`PROBE` Spécifie l'état de détection de défaillance basée sur sonde pour les interfaces configurées avec une adresse de test, à savoir :

- `ok` : indique que la sonde est fonctionnelle et active.
- `failed` : indique que la détection de défaillance basée sur sonde a détecté que l'interface n'est pas opérationnelle.
- `unknown` : indique qu'aucune cible adéquate n'a été trouvée. Par conséquent, les sondes ne peuvent pas être envoyées.
- `disabled` : indique qu'aucune adresse de test IPMP n'est configurée sur l'interface. Par conséquent, la détection de défaillance basée sur sonde est désactivée.

`STATE` Spécifie l'état global de l'interface, comme suit :

- `ok` : indique que l'interface est en ligne et fonctionne normalement sur la base de la configuration des méthodes de détection de défaillance.
- `failed` : indique que l'interface ne fonctionne pas soit parce que la liaison de l'interface est hors service, soit parce que la détection par sonde a déterminé que l'interface n'est pas en mesure d'envoyer ou de recevoir le trafic.
- `offline` : indique que l'interface n'est pas disponible pour utilisation. En règle générale, l'interface est mise hors ligne dans les circonstances suivantes :
  - L'interface est en cours de test.
  - La reconfiguration dynamique est en cours d'exécution.

- L'interface partage une adresse matérielle double avec une autre interface.

- `unknown` : indique que l'état de l'interface IPMP ne peut pas être déterminé car aucune cible de sonde n'a été trouvée pour procéder à la détection de défaillance basée sur sonde.

EXEMPLE 3-12 Obtention d'informations sur les cibles de sonde IPMP

L'option `-t` permet d'identifier les cibles de tests associés à chaque interface d'un groupe IPMP. Le résultat de cette opération dans l'exemple suivant illustre une configuration IPMP qui utilise des adresses de test IPMP pour la détection de défaillance basée sur sonde.

```
% ipmpstat -nt
INTERFACE   MODE          TESTADDR        TARGETS
net0        routes        192.168.85.30   192.168.85.1 192.168.85.3
net1        disabled      --              --
net3        disabled      --              --
net4        routes        192.1.2.200      192.1.2.1
net2        multicast     192.168.10.200   192.168.10.1 192.168.10.2
net6        multicast     192.168.10.201   192.168.10.2 192.168.10.1
net5        multicast     192.168.10.202   192.168.10.1 192.168.10.2
net7        multicast     192.168.10.203   192.168.10.1 192.168.10.2
```

La deuxième sortie illustre une configuration IPMP qui utilise des sondes transitives ou bien la détection de défaillance basée sur sonde sans adresses de test.

```
% ipmpstat -nt
INTERFACE   MODE         TESTADDR         TARGETS
net3        transitive   <net1>           <net1> <net2> <net3>
net2        transitive   <net1>           <net1> <net2> <net3>
net1        routes       172.16.30.100    172.16.30.1
```

Les champs figurant dans la sortie indiquent les informations suivantes.

`INTERFACE` Spécifie chacune des interfaces sous-jacentes d'un groupe IPMP.

`MODE` Spécifie la méthode d'obtention des cibles de sonde.

- `routes` : indique que la table de routage du système est utilisée pour rechercher des cibles de sondes.
- `mcast` : indique que des sondes ICMP multidiffusion sont utilisées pour rechercher des cibles.
- `disabled` : indique que la détection de défaillance basée sur sonde a été désactivée pour l'interface.
- `transitive` : indique qu'une sonde transitive est utilisée pour détecter les défaillances, comme indiqué dans le deuxième exemple. Notez que vous ne pouvez en aucun cas implémenter la détection de défaillance basée sur sonde en utilisant simultanément des sondes

transitives et des adresses de test. Si vous ne souhaitez pas utiliser d'adresses de test, il faut activer les sondes transitives. Si vous ne souhaitez pas utiliser le test transitif, vous devez configurer des adresses test. Pour obtenir une présentation générale, reportez-vous à la section “Détection de défaillance basée sur sonde” à la page 64.

`TESTADDR` Spécifie le nom d'hôte ou, si l'option `-n` est utilisée avec l'option `-t`, l'adresse IP attribuée à l'interface pour envoyer et recevoir les sondes.

Si le test transitif est utilisé, le nom de l'interface fait référence aux interfaces IP sous-jacentes qui ne sont pas activement utilisées pour recevoir des données. Les noms indiquent également que les sondes transitives sont envoyées avec l'adresse source des interfaces spécifiées. Pour que les interfaces IP sous-jacentes actives reçoivent des données, une adresse IP est affichée pour indiquer l'adresse source de sondes ICMP sortantes.

Remarque - Si une interface IP est configurée avec des adresses de test IPv4 et IPv6, les informations de cible de sonde sont affichées séparément pour chaque adresse de test.

`TARGETS` Répertorie les cibles de sondes actuelles dans une liste séparée par des espaces. Les cibles de sonde sont affichées sous la forme de nom d'hôte ou d'adresse IP. Si l'option `-n` est utilisée avec l'option `-t`, les adresses IP sont affichées.

EXEMPLE 3-13 Observation des tests IPMP

L'option `-p` permet d'observer les tests en cours. Lorsque vous ajoutez cette option à la commande `ipmpstat`, des informations sur l'activité des sondes sur le système sont constamment affichées jusqu'à ce que vous mettiez fin à l'opération en appuyant sur les touches Ctrl+C. Vous devez assumer le rôle `root` ou disposer des privilèges appropriés pour exécuter cette commande.

La première sortie illustre une configuration IPMP qui utilise des adresses de test dans le cadre de la détection de défaillance basée sur sonde.

```
# ipmpstat -pn
TIME    INTERFACE   PROBE     NETRTT   RTT      RTTAVG     TARGET
0.11s   net0        589       0.51ms   0.76ms   0.76ms     192.168.85.1
0.17s   net4        612       --       --       --         192.1.2.1
0.25s   net2        602       0.61ms   1.10ms   1.10ms     192.168.10.1
0.26s   net6        602       --       --       --         192.168.10.2
0.25s   net5        601       0.62ms   1.20ms   1.00ms     192.168.10.1
0.26s   net7        603       0.79ms   1.11ms   1.10ms     192.168.10.1
1.66s   net4        613       --       --       --         192.1.2.1
1.70s   net0        603       0.63ms   1.10ms   1.10ms     192.168.85.3
^C
```

La deuxième sortie illustre une configuration IPMP qui utilise des sondes transitives ou bien la détection de défaillance basée sur sonde sans adresses de test.

```
# ipmpstat -pn
TIME    INTERFACE   PROBE      NETRTT   RTT      RTTAVG     TARGET
1.39S   net4        t28        1.05ms   1.06ms   1.15ms     <net1>
1.39s   net1        i29        1.00ms   1.42ms   1.48ms     172.16.30.1
^C
```

Les champs figurant dans la sortie indiquent les informations suivantes.

`TIME` Spécifie l'heure à laquelle une sonde a été envoyé par rapport à l'heure à laquelle la commande `ipmpstat` a été émise. Si une sonde a été lancée avant le démarrage de la commande `ipmpstat`, l'heure s'affiche avec une valeur négative, par rapport au moment de l'émission de la commande.

`INTERFACE` Spécifie l'interface sur laquelle la sonde est envoyée.

`PROBE` Spécifie l'identificateur qui représente la sonde. Si le test transitif est utilisé pour détecter les défaillances, l'identificateur est précédé de `t` pour les tests transitifs ou `i` pour les sondes ICMP.

`NETRTT` Spécifie le temps total d'aller-retour réseau de la sonde (en millisecondes). `NETRTT` couvre le temps entre le moment où le module IP envoie la sonde et le moment où le module IP reçoit les paquets `ack` à partir de la cible. Si le démon `in.mpathd` a déterminé que la sonde est perdue, le champ est vide.

`RTT` Spécifie le temps total d'aller-retour réseau de la sonde (en millisecondes). `RTT` couvre l'intervalle de temps entre le moment où le démon `in.mpathd` exécute le code pour envoyer la sonde et le moment où il termine le traitement des paquets `ack` à partir de la cible. Si le démon a déterminé que la sonde est perdue, le champ est vide. Les pics qui se produisent dans l'intervalle `RTT` mais qui ne figurent pas dans `NETRTT` peuvent indiquer une surcharge du système local.

`RTTAVG` Spécifie le temps moyen d'aller-retour réseau de la sonde sur l'interface entre le système local et la cible. Le temps moyen d'aller-retour permet d'identifier les cibles lentes. Si les données sont insuffisantes pour calculer la moyenne, ce champ est vide.

`TARGET` Permet d'indiquer le nom de l'hôte. Spécifie le nom d'hôte ou, si l'option `-n` est utilisée avec l'option-`p`, l'adresse cible à laquelle la sonde est envoyée.

Personnalisation des résultats de la commande `ipmpstat`

L'-option o permet de personnaliser la sortie de la commande. `ipmpstat` Cette option (utilisée conjointement avec d'autres options `ipmpstat`) permet de sélectionner les champs spécifiques à afficher parmi tous les champs que l'option principale affiche normalement.

Par exemple, l'option `-g` fournit les informations suivantes :

- Groupe IPMP
- nom de groupe IPMP
- Statut du groupe
- Temps de détection de défaillance
- Interfaces sous-jacentes du groupe IPMP

Supposons que vous souhaitiez afficher uniquement le statut des groupes IPMP sur le système. Il faudrait combiner les options `-o` et `-g`, puis spécifier les champs `groupname` et `state`, comme indiqué dans l'exemple suivant :

```
% ipmpstat -g -o groupname,state
GROUPNAME  STATE
ipmp0      ok
accgt1     failed
field2     degraded
```

Pour afficher tous les champs de la commande `ipmpstat` pour un type spécifique d'informations, incluez `-oall` dans la syntaxe.

Utilisation de la commande `ipmpstat` dans des scripts

L'option `-o` s'avère utile lorsque vous exécutez la commande `ipmpstat` d' un script ou à l'aide d'un alias de commande, en particulier si vous voulez également générer une sortie analysable.

Pour générer des informations analysables, il faut combiner les options `-P` et `-o` avec l'une des autres options principales `ipmpstat` principales, ainsi que les champs spécifiques que vous souhaitez afficher.

La sortie analysable diffère de la sortie normale en ce sens que :

- Les en-têtes de colonne sont omis.
- Les champs sont séparés par le signe deux-points (`:`).

- Les champs comportant des valeurs vides sont vides, et non identifiés par un double tiret (`--`).
- Lorsque plusieurs champs sont demandés, si un champ contient le signe de ponctuation deux points (`:`) ou barre oblique inverse (\). Vous pouvez éviter ou exclure ces caractères suivants en ajoutant un préfixe (\).

Pour utiliser correctement la syntaxe `ipmpstat -P`, respectez les règles suivantes :

- Utilisez l'option `-o` *option field* avec l'option `-P`. Séparez les champs d'option par une virgule.
- N'utilisez jamais l'option `-o all` avec l'option `-P`.

Attention - Le non-respect d'un de ces règles entraîne l'échec de la commande `ipmpstat -P`.

L'exemple suivant montre la syntaxe correcte pour à l'aide de l'option `-P` :

```
% ipmpstat -P -o -g groupname,fdt,interfaces
ipmp0:10.00s:net0 net1
acctg1::[net3 net4]
field2:20.00s:net2 net7 (net5) [net6]
```

Le nom de groupe, le temps de détection de défaillance et les interfaces sous-jacentes sont des champs d'informations de groupe. Ainsi, vous utilisez les options `-o` et `-g` avec l'option `-P`.

L'option `-P` est destinée à l'utilisation dans les scripts. L'exemple ci-dessous illustre comment exécuter la commande à partir d'un script`ipmpstat`. Le script affiche le temps de détection de défaillance pour un groupe IPMP.

```
getfdt() {
ipmpstat -gP -o group,fdt | while IFS=: read group fdt; do
[[ "$group" = "$1" ]] && { echo "$fdt"; return; }
done
}
```

♦ ♦ ♦ CHAPITRE 4

Administration de tunnels sur IP

Ce chapitre offre une présentation générale des tâches d'administration dans Oracle Solaris de tunnel IP. Pour plus d'informations sur les tâches, reportez-vous au Chapitre 5, Administration des tunnels IP

Ce chapitre aborde les sujets suivants :

Nouveautés d'administration de tunnels IP

Administration de tunnels IP a été révisée ; désormais, ils sont cohérents avec le nouveau modèle utilisé pour des liaisons de données réseau de l'administration d'Oracle Solaris 11. Dans cette version, vous créez et configurez les tunnels IP à l'aide de la commande `dladm`. Les tunnels peuvent également utiliser d'autres fonctionnalités de liaison de données prises en charge dans cette version. Par exemple, la prise en charge des noms choisis par l'administrateur vous permet d'affecter des tunnels attribuer des noms plus descriptifs. Pour plus d'informations, reportez-vous à la page de manuel `dladm`(1M).

Récapitulatif des fonctionnalités IP Tunnel

Les tunnels IP (également appelée simplement "tunnels dans ce guide) constituent un moyen pour le transport des paquets de données entre différents domaines lorsque le protocole de ces domaines n'est pas pris en charge par les réseaux intermédiaires. Par exemple, avec l'introduction du protocole IPv6, les réseaux IPv6 nécessitent un moyen de communiquer en dehors de leurs frontières dans un environnement où la plupart des réseaux utilisent le protocole IPv4. Cette communication est possible avec l'utilisation des tunnels. Les tunnels IP fournissent une liaison virtuelle entre deux noeuds atteignables en utilisant IP. La liaison peut donc être utilisée pour le transport de paquets IPv6 au sein des réseaux IPv4 afin de permettre la communication IPv6 entre les deux sites IPv6.

Types de tunnels

La mise sous tunnel implique d'encapsuler un paquet IP dans un autre paquet. Cette encapsulation permet au paquet d'atteindre sa destination par le biais de réseaux intermédiaires qui ne prennent pas en charge le protocole du paquet. Les tunnels diffèrent en fonction du type d'encapsulation de paquet.

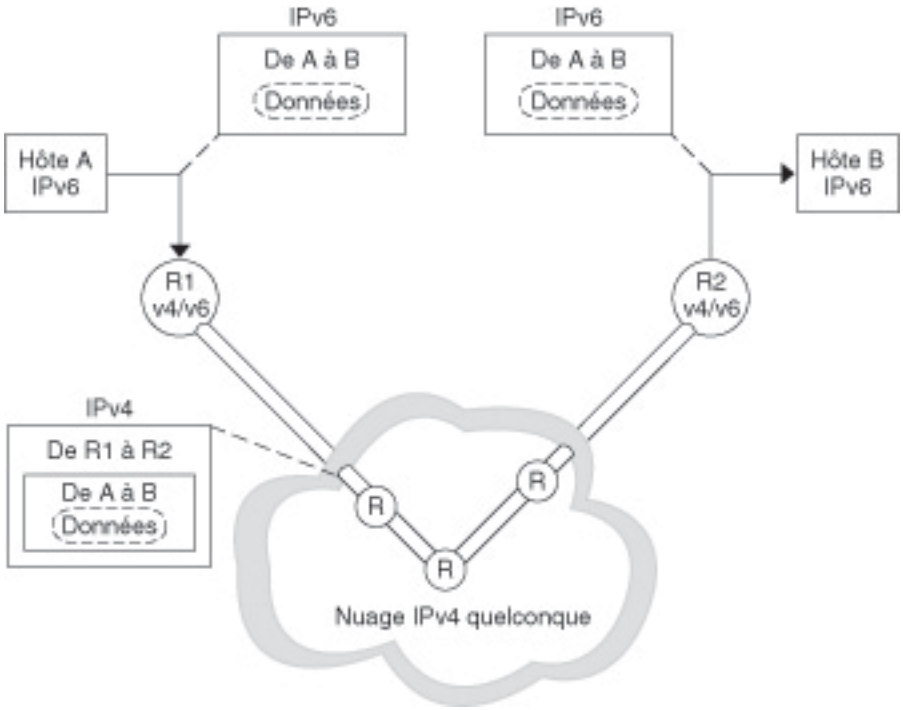
Les types de tunnels suivants sont pris en charge dans Oracle Solaris :

- **tunnels IPv4** – IPv4 les packages sont encapsulés dans un en-tête IPv4 et envoyés à une destination monodiffusion IPv4 préconfigurée. Pour indiquer de façon plus spécifique les paquets acheminés dans le tunnel, les tunnels IPv4 sont également appelés *tunnels IPv4 sur IPv4* ou *tunnels IPv6 sur IPv4*.
- **Tunnels 6to4** : les paquets IPv6 sont encapsulés dans un en-tête IPv4, puis envoyés à une destination IPv4 déterminée automatiquement en fonction du nombre de paquets. La détermination est basée sur un algorithme défini dans le protocole *6to4 protocol*.
- Tunnels **IPv6 :**les paquets IPv6 sont encapsulés dans un en-tête IPv4 et envoyés à une destination IPv4 déterminée automatiquement en fonction du nombre de paquets. La détermination est basée sur un algorithme défini dans le protocole 6to4.

Tunnels dans les environnements réseau combinant IPv6 et IPv4

Un grand nombre de sites dotés de domaines IPv6 peuvent avoir besoin de communiquer avec les autres domaines IPv6 en traversant des réseaux IPv4 IPv6 au cours des premières phases de déploiement. La figure suivante illustre le mécanisme de mise en tunnel (indiqué par "R" dans la figure) entre deux hôtes IPv6 via des routeurs IPv4.

FIGURE 4-1 Mécanisme de mise en tunnel IPv6



Dans la figure, le tunnel se compose de deux routeurs configurés afin de disposer d'une liaison virtuelle point à point entre les deux routeurs sur le réseau IPv4.

Un paquet IPv6 est encapsulé dans un paquet IPv4. Le routeur de bordure du réseau IPv6 configure un tunnel point à point sur plusieurs réseaux IPv4 jusqu'au routeur de bordure du réseau IPv6 de destination. Le paquet est transporté dans le tunnel au routeur de bordure de destination, où le paquet est décapsulé. Le routeur transmet ensuite le paquet IPv6 distinct au noeud de destination.

Tunnels 6to4

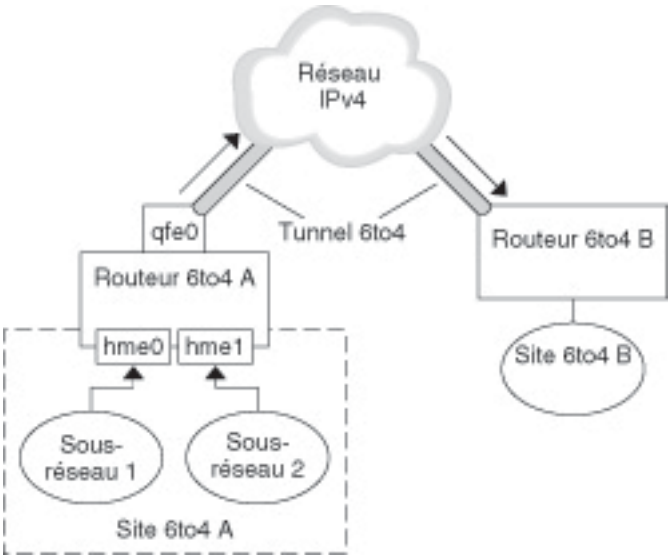
Dans Oracle Solaris, les tunnels 6to4 constituent la méthode temporaire recommandée pour effectuer la transition entre les adressages. Les tunnels *6to4 tunnels* permettent aux sites IPv6 isolés de communiquer, par le biais d'un tunnel automatique, avec un réseau IPv4 ne prenant pas en charge le protocole IPv6. Pour utiliser des tunnels 6to4, vous devez configurer un routeur de bordure sur le réseau IPv6 en tant que point d'extrémité du tunnel 6to4 automatique. Par la

suite, le routeur 6to4 peut participer à un tunnel vers un autre site 6to4 ou vers un site IPv6 natif et non-6to4, le cas échéant.

Topologie d'un tunnel 6to4

Un tunnel 6to4 offre une connectivité IPv6 à tous les sites 6to4 ailleurs. De même, le tunnel offre un lien à l'ensemble des sites IPv6, notamment l'Internet IPv6 natif, à condition d'être configuré pour la transmission vers un routeur relais. La figure suivante illustre un tunnel 6to4 connectant des sites 6to4.

FIGURE 4-2 Tunnel entre deux sites 6to4



La figure représente deux réseaux 6to4 isolés, le site A et le site B. Chaque site a configuré un routeur avec une connexion externe à un réseau IPv4. Un tunnel 6to4 à l'échelle du réseau IPv4 offre une connexion entre sites 6to4.

Pour convertir un site IPv6 en site 6to4, vous devez configurer au moins une interface de routeur prenant en charge 6to4. Cette interface doit assurer la connexion externe au réseau IPv4. Sur cette figure, l'interface du routeur A `net0`connecte le site A au réseau IPv4. L'adresse que vous configurez sur `net0` doit être globale et unique. Vous devez configurer l'interface `net0` avec une adresse IPv4 pour que vous pouvez configurer une interface de tunnel pour prendre en charge 6to4 sur le routeur.

Dans la figure, le site 6to4 A est composé de deux sous-réseaux qui sont connectés aux interfaces `net1` et `net2` du routeur A. Tous les hôtes IPv6 sur l'un des sous-réseaux du site A sont reconfigurés automatiquement avec des adresses dérivées de 6to4 une fois la publication du routeur A reçue.

Le site B est un autre site 6to4 isolé. Pour recevoir correctement le trafic du site A, un routeur de bordure sur le site B doit être configuré pour prendre en charge 6to4. Dans le cas contraire, le routeur ne reconnaît pas les paquets reçus du site A et les abandonne.

Commande `6to4relay`

l'établissement de tunnels 6to4 permet la communication entre les sites 6to4 isolés. Cependant, pour transférer des paquets vers un site IPv6 natif et non-6to4, le routeur 6to4 doit être relié au routeur relais 6to4 par un tunnel. Le *routeur relais 6to4* transfère ensuite les paquets 6to4 au réseau IPv6 et, finalement, au site IPv6 natif. Si un site 6to4 doit échanger des données avec un site IPv6, vous pouvez créer le tunnel approprié à l'aide de la commande `6to4relay`.

Remarque - La liaison de tunnels à des routeurs relais est désactivée car l'utilisation des routeurs relais n'est pas sécurisée. Avant de relier un tunnel à un routeur relais 6to4, vous devez être conscient des problèmes qui peuvent survenir avec ce type de scénario. Pour obtenir des informations détaillées, reportez-vous à la section “Informations importantes pour la création de tunnels vers un routeur relais 6to4” à la page 106 Pour activer la prise en charge d'un routeur relais 6to4, vous pouvez suivre la procédure indiquée à la section “Création et configuration d'un tunnel IP” à la page 112.

Pour plus d'informations, reportez-vous à la page de manuel `6to4`(7M).

Description du flux de paquets dans un tunnel 6to4

Cette section décrit le flux de paquets allant d'un hôte sur un site 6to4 à un autre hôte sur un site 6to4 distant. Ce scénario nécessite la topologie illustrée sur la Figure 4-2, “Tunnel entre deux sites 6to4”. Cela suppose également de configurer au préalable les routeurs et les hôtes 6to4.

La procédure d'authentification est la suivante :

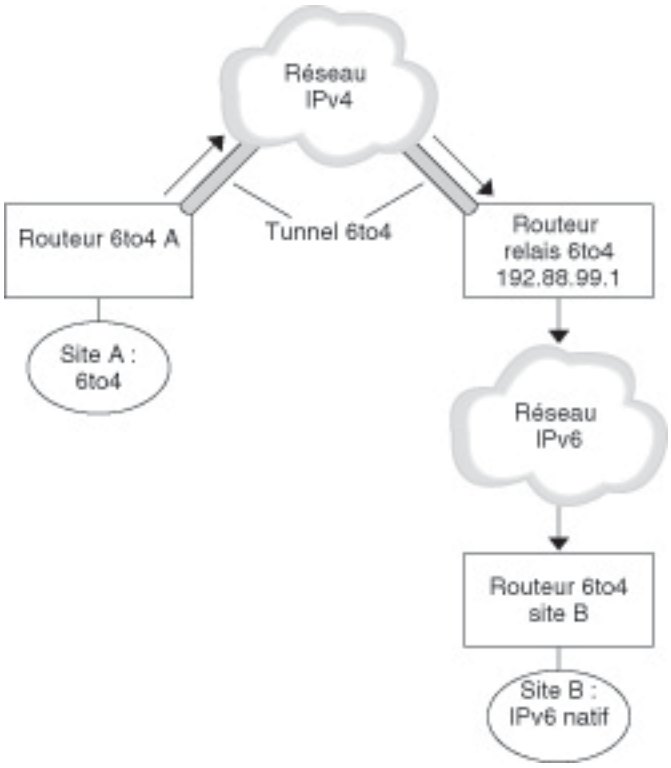
1. Un hôte du sous-réseau 1 appartenant au site 6to4 A envoie une transmission à un hôte du site 6to4 B. Chaque en-tête de paquet possède des adresses 6to4 dérivées source et cible.
2. Le routeur du site A encapsule chaque paquet 6to4 dans un en-tête IPv4. Dans ce processus, le routeur définit l'adresse cible IPv4 de l'en-tête d'encapsulation sur l'adresse du routeur du site B. L'adresse cible IPv6 de chaque paquet IPv6 transmis via l'interface du tunnel contient également l'adresse cible IPv4. Ainsi, le routeur est en mesure de déterminer l'adresse cible IPv4 définie sur l'en-tête d'encapsulation. Ensuite, il utilise la procédure de routage IPv4 standard pour transmettre le paquet sur le réseau IPv4.

3. Tout routeur IPv4 rencontré par les paquets utilise l'adresse IPv4 cible de ces derniers pour la transmission. Cette adresse constitue l'adresse IPv4 globale et unique de l'interface du routeur B, qui sert également de pseudointerface 6to4.
4. Les paquets du site A arrivent sur le routeur B qui les décapsule en paquets IPv6 à partir de l'en-tête IPv4.
5. Le routeur B se sert alors de l'adresse cible des paquets IPv6 pour transmettre ces derniers à l'hôte destinataire sur le site B.

Informations importantes pour la création de tunnels vers un routeur relais 6to4

Les routeurs relais 6to4 fonctionnent en tant que points d'extrémité des tunnels reliant des routeurs 6to4 à des réseaux IPv6 natifs, non 6to4. Les routeurs relais constituent essentiellement des ponts entre le site 6to4 et les sites IPv6 natifs. Ce type de routeur risque de ne pas garantir la sécurité du réseau, c'est pourquoi il ne prend pas en charge le support du routeur relais 6to4. Cependant, si votre site nécessite un tel tunnel, vous pouvez exécuter la commande `6to4relay` pour créer le type de tunnel suivant.

FIGURE 4-3 Tunnel entre un site 6to4 et un routeur relais 6to4



Dans la Figure 4-3, “Tunnel entre un site 6to4 et un routeur relais 6to4”, le site 6to4 A doit communiquer avec un noeud du site natif IPv6 B. La figure indique le chemin du trafic en provenance du site A dans un tunnel 6to4 sur un réseau IPv4. Le tunnel dispose d'un routeur A 6to4 et d'un routeur relais 6to4 à chaque extrémité. Au-delà du routeur 6to4 se trouve le réseau IPv6 auquel le site B IPv6 est connecté.

Flux de paquets entre un site 6to4 et un site IPv6 natif

Cette section décrit le flux de paquets allant d'un hôte sur un site 6to4 à un site IPv6 natif. Ce scénario nécessite la topologie illustrée sur la Figure 4-3, “Tunnel entre un site 6to4 et un routeur relais 6to4”.

La procédure d'authentification est la suivante :

1. Un hôte sur le site 6to4 A envoie une transmission spécifiant un hôte sur le site natif IPv6 B en tant que destination. Chaque en-tête de paquet dispose d'une adresse dérivée de 6to4 en tant qu'adresse source. L'adresse de destination correspond à une adresse IPv6 standard.
2. Le routeur 6to4 du site A encapsule chaque paquet dans un en-tête IPv4, dont la destination correspond à l'adresse IPv4 du routeur relais 6to4. Ensuite, il utilise la procédure de routage IPv4 standard pour transmettre le paquet sur le réseau IPv4. Tout routeur IPv4 rencontré par les paquets envoie ceux-ci vers le routeur relais 6to4.
3. Le routeur relais 6to4 anycast le plus proche physiquement du site A récupère les paquets destinés au groupe anycast `192.88.99.1`.

Remarque - Les routeurs relais 6to4 faisant partie du groupe anycast de routeurs relais 6to4 possèdent l'adresse IP `192.88.99.1`. Cette adresse anycast constitue l'adresse par défaut des routeurs relais 6to4. Si vous avez besoin d'un routeur relais 6to4 spécifique, vous pouvez supprimer celui par défaut et spécifier l'adresse IPv4 du routeur en question.

---

4. Ce routeur relais décapsule ensuite l'en-tête IPv4 des paquets 6to4, dévoilant l'adresse de destination sur le réseau IPv6.
5. Le routeur relais envoie ensuite les paquets qui sont à présent IPv6 uniquement sur le réseau IPv6, où ils seront ensuite récupérés par un routeur sur le site B. Le routeur transmet ensuite les paquets au noeud de destination IPv6.

A propos du déploiement IP Tunnels

Pour déployer les tunnels IP correctement, vous devez effectuer deux tâches principales. En premier lieu, créez la liaison de tunnel IP interface, veillez à configurer un sur le tunnel. Vous trouverez ci-dessous les contraintes pour la création des tunnels et de leurs interfaces IP correspondantes.

Exigences de création de tunnels IP

Pour créer des tunnels correctement, vous devez remplir les exigences suivantes :

- Si vous utilisez des noms d'hôte plutôt que des adresses IP littérales, ces noms doivent être résolus en adresses IP valides compatibles avec le type de tunnel.
- Le tunnel IPv4 ou IPv6 que vous créez ne doit pas partager les mêmes adresses source et de destination de tunnel avec un autre tunnel configuré.
- Le tunnel IPv4 ou IPv6 que vous créez ne doit pas partager la même adresses source avec un tunnel 6to4 existant.
- Si vous créez un tunnel 6to4, celui-ci ne doit pas partager la même adresse source avec un autre tunnel configuré.

Pour plus d'informations, reportez-vous à la“ Planification de l’utilisation de tunnels dans le réseau ” du manuel “ Planification du développement du réseau dans Oracle Solaris 11.2 ”.

Exigences relatives aux tunnels et aux interfaces IP

Chaque type de tunnel est doté d'exigences spécifiques en matière d'adresses IP sur l'interface IP que vous configurez sur le tunnel. Les exigences sont résumées dans le tableau ci-dessous.

TABLEAU 4-1 Exigences en matière de tunnels et d'interface IP

Type de tunnel	Interface IP autorisée sur le tunnel	Exigence d'interface IP
Tunnel IPv4	Interface IPv4	Les adresses locales et distantes sont spécifiées manuellement.
	Interface IPv6	Les adresses locales et distantes de liaison locale sont définies automatiquement lors de l'exécution de la commande `ipadm create-addr -T addrconf`. Voir la page de manuel `ipadm`(1M).
Tunnel IPv6	Interface IPv4	Les adresses locales et distantes sont spécifiées manuellement.
	Interface IPv6	Les adresses locales et distantes de liaison locale sont définies automatiquement lors de l'exécution de la commande `ipadm create-addr -T addrconf`. Voir la page de manuel `ipadm`(1M).
Tunnel 6to4	Interface IPv6 uniquement	L'adresse IPv6 par défaut est sélectionnée automatiquement lors de l'exécution de la commande `ipadm create-ip`. Voir la page de manuel `ipadm`(1M).

Vous pouvez remplacer les adresses de liaison locale définies automatiquement pour les interfaces IPv6 sur IPv4 Tunnels IPv6, ou en spécifiant une `interface-id` locale et distante à l'aide de la commande `ipadm create-addr -T addrconf`.

♦ ♦ ♦ CHAPITRE 5

Administration des tunnels IP

Ce chapitre décrit les tâches de tunnels dans Oracle Solaris IP d'administration. Pour obtenir un aperçu de l'administration de tunnelss dans Oracle Solaris, reportez-vous au Chapitre 4, Administration de tunnels sur IP.

Ce chapitre aborde les sujets suivants :

Administration de tunnels IP dans Oracle Solaris

Dans cette version Oracle Solaris à partir de l'administration de tunnel est séparés par des virgules, de la configuration de l'interface IP. Vous administrez l'aspect de la liaison de données des tunnels IP avec la commande`dladm` et l'aspect de la configuration IP (y compris ceux des tunnels IP) à l'aide de la commande `ipadm`.

Les sous-commandes `dladm` suivantes permettent de configurer les tunnels IP :

- `create-iptun`
- `modify-iptun`
- `show-iptun`
- `delete-iptun`
- `set-linkprop`

Pour plus d'informations, reportez-vous à la page de manuel `dladm`(1M).

Remarque - l'administration de tunnels IP est étroitement liée à la configuration avec IPsec. Par exemple, les VPN IPsec sont l'une des utilisations principales de la mise sous tunnel IP. Pour plus d'informations sur la sécurité dans Oracle Solaris, reportez-vous au Chapitre 6, “ A propos de l’architecture de sécurité IP ” du manuel “ Sécurisation du réseau dans Oracle Solaris 11.2 ”. Pour plus d'informations sur la configuration des protocoles IPsec, reportez-vous au Chapitre 7, “ Configuration d’IPsec ” du manuel “ Sécurisation du réseau dans Oracle Solaris 11.2 ”.

Administration des tunnels IP

Cette section s'articule autour des rubriques suivantes :

▼ Création et configuration d'un tunnel IP

1. **Connectez-vous en tant qu'utilisateur `root`.**

2. **Créez le tunnel.**

```
# dladm create-iptun [-t] -T type -a [local|remote]=addr,... tunnel-link
```

`-t`	Crée un tunnel temporaire. Par défaut, la commande crée un tunnel persistant. Si vous souhaitez configurer une interface IP sur le tunnel, vous devez créer un tunnel persistant et ne pas utiliser l'option `-t`.
`-T` *type*	Spécifie le type de tunnel à créer. Cet argument est requis pour créer tous les types de tunnel.
`-a [local|remote]=`*address,...*	Spécifie les adresses IP littérales ou les noms d'hôte correspondant aux adresses locales et à l'adresse de tunnel distant. Les adresses doivent être valides et déjà créées dans le système. Suivant le type de tunnel, spécifiez soit une seule adresse, soit les adresses locales et distantes. Si vous spécifiez les adresses locales et distantes, vous devez les séparer à l'aide d'une virgule. ■ Les tunnels `IPv4` nécessitent des adresses IPv4 locales et distantes pour fonctionner. ■ Les tunnels `IPv6` nécessitent des adresses IPv6 locales et distantes pour fonctionner. ■ Les tunnels `6to4` nécessitent une adresse IPv4 locale pour fonctionner.

Remarque - Pour les configurations de liaison de données de tunnel IP, si vous utilisez des noms d'hôte en guise d'adresses, ces noms d'hôte sont enregistrés dans le stockage de configuration. Lors d'une initialisation ultérieure du système, si la résolution de noms donne des adresses IP différentes de celles utilisées lors de la création du tunnel, ce dernier acquiert une nouvelle configuration.

tunnel-link Spécifie la liaison de tunnel IP. Avec la prise en charge des noms significatifs dans une administration réseau-liaison, les noms de tunnel ne sont plus limités au type de tunnel que vous créez. Au lieu de cela, vous pouvez affecter tout nom choisi par l'administrateur vers un tunnel. Les noms de tunnel se composent d'une chaîne et du numéro de point de connexion physique, par exemple, *montunnel0*. Pour connaître les règles régissant l'attribution de noms explicites, reportez-vous à " Règles d'affectation de noms de liaison valides " du manuel " Configuration et administration des composants réseau dans Oracle Solaris 11.2 ".

3. (Facultatif) Définissez les valeurs de la limite de saut ou de la limite d'encapsulation.

```
# dladm set-linkprop -p [hoplimit=value] [encaplimit=value] tunnel-link
```

`hoplimit` Spécifie la limite de saut de l'interface de tunnel pour la mise en tunnel sur IPv6. *hoplimit* est l'équivalent du champ IPv4 de durée de vie pour la mise en tunnel sur IPv4.

`encaplimit` Spécifie le nombre de niveaux de tunnels imbriqués autorisés pour un paquet. Cette option s'applique uniquement aux tunnels IPv6.

Les valeurs définies pour `hoplimit` et `encaplimit` doivent être comprises dans une plage acceptable. `hoplimit` et `encaplimit` sont des propriétés de liaison de tunnel. Par conséquent, ces propriétés sont administrées par les mêmes sous-commandes `dladm` que pour les autres propriétés de liaison. Les sous-commandes que vous utilisez sont `dladm set-linkprop`, `dladm reset-linkprop` et `dladm show-linkprop`.

4. Créez une interface IP sur le tunnel.

```
# ipadm create-ip tunnel-interface
```

où *tunnel-interface* utilise le même nom que la liaison de tunnel.

5. Assignez des adresses IP locales et distantes à l'interface de tunnel.

```
# ipadm create-addr [-t] -a local=address,remote=address interface
```

où *interface* spécifie l'interface de tunnel.

Pour plus d'informations, reportez-vous à la page de manuel`ipadm`(1M) et “ Configuration et administration des composants réseau dans Oracle Solaris 11.2 ”.

6. **(Facultatif) Vérifiez le statut de la configuration d'interface de l'IP du tunnel.**

```
# ipadm show-addr interface
```

Exemple 5-1 Création d'une interface IPv6 sur un tunnel IPv4

L'exemple suivant montre comment créer un tunnel IPv6 via IPv4 permanent.

```
# dladm create-iptun -T ipv4 -a local=192.0.2.23,remote=203.0.113.14 private0
# dladm set-linkprop -p hoplimit=200 private0
# ipadm create-ip private0
# ipadm create-addr -T addrconf private0
private0/v6
# ipadm show-addr private0/
ADDROBJ           TYPE     STATE        ADDR
private0/v6       addrconf ok fe80::c000:217->fe80::cb00:710e
```

Pour ajouter d'autres adresses, respectez la même syntaxe. Par exemple, vous pouvez ajouter une adresse globale comme suit :

```
# ipadm create-addr -a local=2001:db8:4728::1,remote=2001:db8:4728::2 private0
private0/v6a
# ipadm show-addr private0/
ADDROBJ           TYPE     STATE        ADDR
private0/v6       addrconf ok fe80::c000:217->fe80::cb00:710e
private0/v6a      static   ok 2001:db8:4728::1->2001:db8:4728::2
```

Notez que le préfixe `2001:db8` de l'adresse IPv6 est un préfixe IPv6 spécial utilisé spécifiquement pour les exemples de documentation.

Exemple 5-2 Création d'une interface IPv4 sur un tunnel IPv4

L'exemple suivant montre comment créer un tunnel IPv4 via IPv4 permanent.

```
# dladm create-iptun -T ipv4 -a local=192.0.2.23,remote=203.0.113.14 vpn0
# ipadm create-ip vpn0
# ipadm create-addr -a local=10.0.0.1,remote=10.0.0.2 vpn0
vpn0/v4
# ipadm show-addr vpn0/
ADDROBJ           TYPE     STATE        ADDR
vpn0/v4           static   ok 10.0.0.1->10.0.0.2
```

Vous pouvez configurer la stratégie IPsec davantage pour fournir des connexions sécurisées aux paquets circulant dans ce tunnel. Pour plus d'informations, reportez-vous au Chapitre 7, “ Configuration d’IPsec ” du manuel “ Sécurisation du réseau dans Oracle Solaris 11.2 ”.

Exemple 5-3 Création d'une interface IPv6 sur un tunnel IPv6

L'exemple suivant montre comment créer un tunnel IPv6 via IPv6 permanent.

```
# dladm create-iptun -T ipv6 -a local=2001:db8:feed::1234,remote=2001:db8:beef::4321 tun0
# ipadm create-ip tun0
# ipadm create-addr -T addrconf tun0
tun0/v6
# ipadm show-addr tun0/
ADDROBJ           TYPE     STATE        ADDR
tun0/v6           addrconf ok fe80::1234->fe80::4321
```

Pour ajouter des adresses comme une adresse globale ou d'autres adresses locales et distantes, utilisez la commande `ipadm` comme suit :

```
# ipadm create-addr -a local=2001:db8:cafe::1,remote=2001:db8:cafe::2 tun0
tun0/v6a
# ipadm show-addr tun0/
ADDROBJ           TYPE     STATE        ADDR
tun0/v6           addrconf ok fe80::1234->fe80::4321
tun0/v6a          static   ok 2001:db8:cafe::1->2001:db8:cafe::2
```

▼ Configuration d'un tunnel 6to4

Dans les tunnels 6to4, un routeur 6to4 doit agir en tant que routeur IPv6 pour les noeuds des sites 6to4 du réseau. Par conséquent, lors de la configuration d'un routeur 6to4, ce routeur doit également être configuré en tant que routeur IPv6 sur ses interfaces physiques. Pour plus d'informations sur la configuration d'un hôte Oracle Solaris en tant que routeur, reportez-vous à “ Configuration d’un routeur IPv6 ” du manuel “ Configuration d’un système Oracle Solaris 11.2 en tant que routeur ou équilibreur de charge ”.

1. Créez un tunnel 6to4.

```
# dladm create-iptun -T 6to4 -a local=address  tunnel-link
```

-a local=*address*
: Spécifie l'adresse locale de tunnel qui doit déjà exister dans le système pour être valide.

tunnel-link
: Spécifie la liaison de tunnel IP. Avec la prise en charge des noms significatifs dans une administration réseau-liaison, les noms de tunnel ne sont plus limités au type de tunnel que vous créez. Au lieu de cela, vous pouvez affecter un tunnel tout nom choisi par l'administrateur. Les noms de tunnel se composent d'une chaîne et du numéro de point de connexion physique, par exemple, *montunnel0*. Pour plus d'informations, reportez-vous à “ Règles d’affectation de noms de liaison valides ” du manuel “ Configuration et administration des composants réseau dans Oracle Solaris 11.2 ”.

2. **Créer l'interface IP de tunnel.**

   ```
   # ipadm create-ip tunnel-interface
   ```

   où *tunnel-interface* utilise le même nom que la liaison de tunnel.

3. **(Facultatif) Ajoutez d'autres adresses IPv6 pour une utilisation par le tunnel.**

4. **Editez le fichier `/etc/inet/ndpd.conf`.**

   ```
   # pfedit /etc/inet/ndpd.conf
   ```

5. **Indiquer routage 6to4 en ajoutant les deux lignes suivantes au fichier.**

   ```
   if subnet-interface AdvSendAdvertisements 1
   IPv6-address subnet-interface
   ```

   où la première ligne spécifie le sous-réseau qui reçoit cette indication, et *subnet-interface* fait référence à la liaison à laquelle est connecté le sous-réseau. Les adresses IPv6 sur la seconde ligne doivent avoir le préfixe 6to4 `2000` qui est utilisé pour les adresses IPv6 dans les tunnels 6to4.

   Pour des informationsplus détaillées concernant le fichier `ndpd.conf`, reportez-vous à la page de manuel `ndpd.conf`(4).

6. **Activez le transfert IPv6.**

   ```
   # ipadm set-prop -p forwarding=on ipv6
   ```

7. **Sélectionnez l'une des options suivantes :**

   - **Redémarrez le routeur.**
   - **Vous pouvez également exécuter la commande `sighup` sur le démon de `/etc/inet/in.ndpd` pour commencer la publication du routeur.**

     Les noeuds IPv6 de chaque sous-réseau devant recevoir le préfixe 6to4 sont alors automatiquement définis sur les nouvelles adresses 6to4 dérivées.

8. **Ajoutez ces nouvelles adresses au service de noms utilisé par le site 6to4.**

   Pour obtenir des instructions, reportez-vous au Chapitre 4, “ Administration des services de noms et d’annuaire sur un client Oracle Solaris ” du manuel “ Configuration et administration des composants réseau dans Oracle Solaris 11.2 ”.

Exemple 5-4 Création de tunnel 6to4

L'exemple suivant illustre comment vous pouvez créer un tunnel 6to4. Notez que seules les interfaces IPv6 peuvent être configurées sur les tunnels 6to4. Dans cet exemple, l'interface de sous-réseau est `net0` à laquelle `/etc/inet/ndpd.conf` fait référence.

```
# dladm create-iptun -T 6to4 -a local=192.0.2.23 tun0
# ipadm create-ip tun0
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           dhcp     ok           192.0.2.23/24
lo0/v6            static   ok           ::1/128
tun0/v6           static   ok           2002:c000:217::1/16

# ipadm create-addr -T addrconf net0
net0/v6
# ipadm create-addr -a 2002:c000:217:cafe::1 net0
net0/v6a
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           dhcp     ok           192.0.2.23/24
lo0/v6            static   ok           ::1/128
net0/v6           addrconf ok     fe80::214:4fff:fef9:b1a9/10
net0/v6a          static   ok           2002:c000:217:cafe::1/64
tun0/v6           static   ok           2002:c000:217::1/16

# vi /etc/inet/ndpd.conf
if net0 AdvSendAdvertisements on
prefix 2002:c000:217:cafe::0/64 net0

# ipadm set-prop -p forwarding=on ipv6
```

Notez que pour les tunnels 6to4, le préfixe de l'adresse IPv6 est `2002`.

▼ Procédure de configuration d'un tunnel 6to4 relié à un routeur relais 6to4

Attention - En raison de problèmes de sécurité importants, la prise en charge du routeur de relais 6to4 est désactivé par défaut en Oracle Solaris. Reportez-vous à " Problèmes de sécurité lors de la création d'un tunnel vers un routeur relais 6to4 " du manuel " Dépannage des problèmes d'administration réseau dans Oracle Solaris 11.2 ".

Avant de commencer

Avant de configurer un tunnel relié à un routeur relais 6to4, vous devez avoir effectué les tâches suivantes :

- Configurer un routeur 6to4 sur votre site. "Création et configuration d'un tunnel IP" à la page 112
- Vérification des problèmes de sécurité susceptibles de se produire avec un tunnel relié à un routeur relais 6to4.

1. **Vous pouvez relier un tunnel à un routeur relais 6to4 de deux façons :**

- **Liaison a un routeur relais 6to4 de type anycast.**

  ```
  # 6to4relay -e
  ```

  L'option `-e` crée un tunnel entre le routeur 6to4 et un routeur de relais anycast 6to4. Les routeurs relais 6to4 anycast possèdent l'adresse IPv4 courante `192.88.99.1`. Le routeur relais anycast le plus proche (physiquement) de votre site devient le point d'extrémité du tunnel 6to4. Ce routeur relais gère ensuite l'envoi des paquets entre votre site 6to4 et un site IPv6 natif.

  Pour obtenir des informations détaillées, reportez-vous àRFC 3068 « Un préfixe Anycast pour un routeur de relais 6to4 » (http://www.rfc-editor.org/rfc/rfc3068.txt).

- **Liaison a un routeur relais 6to4 de type spécifique.**

  ```
  # 6to4relay -e -a relay-router-address
  ```

  L'option `-a` est toujours suivie d'une adresse de routeur spécifique. Remplacez *relay-router-address* par l'adresse IPv4 du routeur relais 6to4 spécifique que vous souhaitez relier au tunnel.

Le tunnel relié au routeur relais 6to4 reste actif pendant la suppression de la pseudointerface du tunnel 6to4.

2. **Supprimez le tunnel relié au routeur relais 6to4 lorsqu'il n'est plus nécessaire.**

   ```
   # 6to4relay -d
   ```

3. **(Facultatif) (Facultatif) Configurez un tunnel au routeur relais 6to4 qui conserve ses paramètres après chaque redémarrage.**

   Si votre site requiert, pour quelque raison qu'il soit, que les paramètres du tunnel relié au routeur relais 6to4 soient redéclarés à chaque redémarrage du routeur, effectuez la procédure suivante :

   a. **Modifiez le fichier`/etc/default/inetinit`.**

      ```
      # pfedit /etc/default/inetinit
      ```

      La ligne à modifier se trouve à la fin du fichier.

   b. **Changez la valeur `NO` dans la ligne `ACCEPT6TO4RELAY=NO` à `YES`.**

   c. **(Facultatif) Créez un tunnel relié à un routeur relais 6to4 spécifique dont les paramètres sont conservés après chaque redémarrage.**

      Pour le paramètre `RELAY6TO4ADDR`, remplacez l'adresse `192.88.99.1` par l'adresse IPv4 du routeur relais 6to4 à utiliser.

Exemple 5-5 Obtention d'informations sur le statut de la prise en charge des routeurs relais 6to4

La commande `/usr/bin/6to4relay` vous permet de savoir si les routeurs relais 6to4 sont pris en charge ou non par votre site. L'exemple suivant présente la sortie obtenue lorsque les routeurs relais 6to4 ne sont pas pris en charge (sortie par défaut d'Oracle Solaris) :

```
# 6to4relay
6to4relay: 6to4 Relay Router communication support is disabled.
```

Lorsque la prise en charge des routeurs relais 6to4 est activée, la sortie suivante s'affiche :

```
# 6to4relay
6to4relay: 6to4 Relay Router communication support is enabled.
IPv4 remote address of Relay Router=192.88.99.1
```

La modification d'une configuration de tunnel IP

Vous pouvez charger la configuration d'un tunnel à l'aide de la syntaxe de commande suivante :

```
# dladm modify-iptun -a [local|remote]=addr,... tunnel-link
```

Vous ne pouvez pas modifier le type d'un tunnel existant. Par conséquent, l'option `-T` *type* n'est pas autorisée pour cette commande. Seuls les paramètres de tunnel suivants peuvent être modifiés :

`-a [local|remote]=`*address,...*

Spécifie les adresses IP littérales ou les noms d'hôte correspondant aux adresses locales et à l'adresse de tunnel distant. Suivant le type de tunnel, spécifiez soit une seule adresse, soit les adresses locales et distantes. Si vous spécifiez les adresses locales et distantes, vous devez les séparer à l'aide d'une virgule.

- Les tunnels `IPv4` nécessitent des adresses IPv4 locales et distantes pour fonctionner.
- Les tunnels `IPv6` nécessitent des adresses IPv6 locales et distantes pour fonctionner.
- Les tunnels `6to4` nécessitent une adresse IPv4 locale pour fonctionner.

Pour les configurations de liaison de données de tunnel IP, si vous utilisez des noms d'hôte en guise d'adresses, ces noms d'hôte sont enregistrés dans le stockage de configuration. Lors d'une initialisation ultérieure du système, si la résolution de noms donne des adresses IP différentes de celles utilisées lors de la création du tunnel, ce dernier acquiert une nouvelle configuration.

Si vous modifiez les adresses locales et distantes du tunnel, assurez-vous que ces adresses sont cohérentes par rapport au type de tunnel que vous modifiez.

- Pour modifier le nom de la liaison de tunnel, utilisez la commande `dladm rename-link` et non la commande `modify-iptun` de la manière suivante :

  `# dladm rename-link` *old-tunnel-link new-tunnel-link*
- Pour modifier les propriétés de tunnel telles que `hoplimit` ou `encaplimit`, utilisez la commande `dladm set-linkprop` et non la commande `modify-iptun`.

EXEMPLE 5-6 Modification de l'adresse et des propriétés d'un tunnel

L'exemple suivant comporte deux procédures. Tout d'abord, les adresses locales et distantes du tunnel IPv4 `vpn0` sont modifiées temporairement. Une fois le système réinitialisé, le tunnel revient à l'utilisation des adresses d'origine. La seconde commande indique comment modifier la valeur `hoplimit` de `vpn0` à 60.

```
# dladm modify-iptun -t -a local=10.8.48.149,remote=192.168.2.3 vpn0

# dladm set-linkprop -p hoplimit=60 vpn0
```

L'affichage d'une configuration de tunnel IP

Vous pouvez afficher la configuration d'un tunnel IP à l'aide de la syntaxe de commande suivante :

```
# dladm show-iptun [-p] -o fields [tunnel-link]
```

`-p`	Affiche les informations dans une format analysable. Cet argument est facultatif.
`-o` *fields*	Affiche les champs sélectionnés qui fournissent des informations spécifiques au tunnel.
tunnel-link	Spécifie le tunnel dont vous souhaitez afficher les informations de configuration. Cet argument est facultatif. Si vous omettez le nom de tunnel, la commande affiche les informations à propos de tous les tunnels du système.

EXEMPLE 5-7 Affichage des informations à propos de tous les tunnels

Dans l'exemple suivant, un seul tunnel existe sur le système.

```
# dladm show-iptun
LINK     TYPE     FLAGS     LOCAL          REMOTE
tun0     6to4     --        192.168.35.10  --
vpn0     ipv4     --        10.8.48.149    192.168.2.3
```

EXEMPLE 5-8 Affichage de champs sélectionnés dans un format analysable

Dans l'exemple ci-dessous , seuls les champs spécifiques comportant des informations sur les tunnels sont affichés.

```
# dladm show-iptun -p -o link,type,local
tun0:6to4:192.168.35.10
vpn0:ipv4:10.8.48.149
```

Propriétés, dans laquelle apparaît le de tunnel IP

Les propriétés de l'affichage à l'aide de la liaison du tunnel la syntaxe de commande suivante :

```
# dladm show-linkprop [-c] [-o fields] [tunnel-link]
```

-c	Affiche les informations dans une format analysable. Cet argument est facultatif.
-o *fields*	Affiche les champs sélectionnés fournissant des informations spécifiques à propos des propriétés du lien.
tunnel-link	Spécifie le tunnel dont vous souhaitez afficher les informations sur les propriétés. Cet argument est facultatif. Si vous omettez le nom de tunnel, la commande affiche les informations à propos de tous les tunnels du système.

EXEMPLE 5-9 Affichage des propriétés d'un tunnel

L'exemple ci-dessous illustre comment afficher toutes les propriétés d'une liaison de tunnel.

```
# dladm show-linkprop iptun0
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
iptun0   autopush        rw   --           --           --        --
iptun0   zone            rw   --           --           --        --
iptun0   state           r-   up           up           up        up,down
iptun0   mtu             rw   1480         1480         1480      1280-1480
iptun0   maxbw           rw   --           --           --        --
iptun0   cpus            rw   --           --           --        --
iptun0   rxfanout        rw   --           8            8         --
iptun0   pool            rw   --           --           --        --
iptun0   priority        rw   medium       medium       medium    low,medium,
                                                                  high
iptun0   hoplimit        rw   64           64           64        1-255
iptun0   protection      rw   --           --           --        mac-nospoof,
                                                                  restricted,
                                                                  ip-nospoof,
                                                                  dhcp-nospoof
iptun0   allowed-ips     rw   --           --           --        --
iptun0   allowed-dhcp-cids rw --           --           --        --
```

```
iptun0   rxrings         rw   --          --          --        --
iptun0   txrings         rw   --          --          --        --
iptun0   txringsavail    r-   0           0           --        --
iptun0   rxringsavail    r-   0           0           --        --
iptun0   rxhwclntavail   r-   0           0           --        --
iptun0   txhwclntavail   r-   0           0           --        --
```

▼ Suppression d'un tunnel IP

1. **Utilisez la syntaxe adéquate pour déconnecter l'interface IP configurée sur le tunnel en fonction du type de l'interface.**

   ```
   # ipadm delete-ip tunnel-link
   ```

   Remarque - Pour supprimer un tunnel correctement, aucune interface IP existante ne peut être montée sur le tunnel.

2. **Supprimez le tunnel IP.**

   ```
   # dladm delete-iptun tunnel-link
   ```

   La seule option pour cette commande est `-t`, laquelle entraîne une suppression temporaire du tunnel. Le tunnel est restauré lors de la réinitialisation du système.

Exemple 5-10 Suppression d'un tunnel IPv6 configuré avec une interface IPv6

Dans cet exemple, un tunnel persistant est supprimé définitivement.

```
# ipadm delete-ip ip6.tun0
# dladm delete-iptun ip6.tun0
```

Indice

E

F

G

H

I

S

T

U

W